New Local Search Methods for Improving the Lagrangian Relaxation-Based Unit Commitment Solution

Guidance

Associate Professor Nobuo YAMASHITA

Takeshi SEKI

2006 Graduate Course

in

Department of Applied Mathematics and Physics

Graduate School of Informatics

Kyoto University

February 2008

Abstract

The unit commitment problem (UCP) for an electric power system is to determine the schedules of power units that minimize the total production cost over a planning horizon while satisfying the load demand, spinning reserve, and operating constraints of individual units. The UCP is formulated as a nonlinear mixed-integer programming problem that includes 0-1 variables representing the on-off states of the units. When the number of units is large and the planning horizon is long, the UCP is a large-scale problem, for which an exact optimal solution is difficult to obtain within a reasonable computation time. Therefore, several methods have been proposed to obtain an approximate solution of the UCP. Among these methods, the Lagrangian relaxation (LR) method is useful for large-scale UCPs. The LR method is first used to solve the dual problem of the UCP and is then used to construct a feasible solution from the dual solution by using some heuristics. However, the quality of the feasible solution is not satisfactory.

In the present paper, we propose new local search methods for improving the feasible solution obtained by the LR method. We define the neighborhood of the local search as the feasible set in which the schedules of all but one or two units are fixed. The neighborhood search can then be executed by solving the one-unit or two-unit commitment problems, which are efficiently solved by dynamic programming. In this search, quadratic programming problems with a particular structure should be solved frequently. Since a general quadratic programming solver does not exploit such a special structure, a great deal of time is required when the number of units is large. Therefore, we also propose a technique for solving the quadratic programming problems efficiently by exploiting the particular structure of the quadratic programming problems.

Numerical results show that the proposed local search methods can find feasible schedules for which the costs are lower than those obtained by the existing methods, which are based on mixed-integer programming or genetic algorithms. The applicability of the proposed methods to long-term UCPs is also demonstrated.

Contents

1 Introduction

Electric power is an essential form of energy in modern life. In many countries and in certain areas, thermal power generation is a major source of power and reducing the production costs is important. The price of electricity is linked with production costs, especially fuel costs. In recent years, steep increases in the price of natural resources such as heavy oil or natural gas have increased the production costs of thermal power plants.

Power generation systems usually consist of several thermal units. The unit commitment problem (UCP) is to determine the schedules and generations that minimize the total production cost over the planning horizon while satisfying the load demand, spinning reserve, and operating constraints of individual units. The UCP is commonly formulated as a nonlinear mixed-integer programming problem with 0-1 variables representing the on-off states of the thermal units. As such, when the number of units is large and the planning horizon is long, the problem is an extremely large-scale problem. The exact optimal solution of such a large-scale UCP is difficult to obtain in a practical computation time.

Several methods have been proposed by which to obtain an approximate solution of the UCP efficiently [13], including priority list methods [14], dynamic programming (DP) [15], branch-and-bound methods (B&B) [9], mixed-integer programming (MIP) [7], and Lagrangian relaxation methods (LR) [2, 11, 12, 8]. The priority list method is the simplest of these methods and is able to obtain a feasible solution within a short computation time, even if the problem size is large. However, the obtained solution is not satisfactory. Although methods such as DP, B&B, and MIP can theoretically obtain an exact (or near exact) solution, they require an impractical computation time for a large-scale UCP. The LR method basically solves the dual problem of the UCP. The objective function of the dual problem is represented as the optimal value of the Lagrangian relaxed problem, which can be decomposed into small subproblems of each unit. Using this characteristic, we can obtain the dual solution of the UCP efficiently even if the problem size is large. However, in general, obtaining the solution through dual optimization is not feasible for the UCP. Therefore, we must find a feasible solution by some heuristics. The main disadvantage of the LR method is that such a feasible solution is often unsatisfactory.

In the present paper, we consider the large-scale UCP, which consists of several units over a long planning horizon. Therefore, we apply the LR-based method. As stated above, the LR-based feasible solution is often disappointing. In order to overcome this disadvantage, we propose new local search methods to improve the feasible solution. Methods to improve the LR-based solution have been proposed in previous studies [16, 12]. However, since the method proposed in [16] uses mixed-integer programming, it is not suitable for large-scale UCPs. In the method proposed in [12], a local search method is proposed in the present study. This method represents the continuous operating states as one block in order to satisfy the minimum up time constraints and replaces the blocks to improve the solution. Since the local search is restricted to individual blocks, the improvement is limited. We therefore propose new local search methods, in which we may modify the schedules of one or two units freely. Therefore, the proposed methods can achieve a wider search than the method of [12]. Therefore, we expect that the methods proposed herein can obtain a better solution. However, expanding the search area of the local search might cause a long computation time. In the present paper, we also propose a technique based on the nonlinear optimization for efficient neighborhood search of the proposed local search methods.

The remainder of the present paper is organized as follows. Section 2 describes the formulation of the common unit commitment problem considered herein. The Lagrangian relaxation procedure for the UCP is discussed in Section 3. Section 4 presents a detailed description of the proposed local search methods. Numerical results are presented and discussed in Section 5. Finally, conclusions are presented in Section 6.

2 Unit Commitment Problem

In this section, we formulate the unit commitment problem (UCP).

2.1 Notation

The following notation will be used herein.

Indices:

i Index for unit.

t Index for time period.

Constants:

I Set of units.

T Total number of time periods.

S_i^{hot} Hot startup cost of unit i.

S_i^{cold} Cold startup cost of unit i.

D_t Load demand in time period t.

R_t Spinning reserve in time period t.

$p_i^{\max}$ Maximum generation of unit i.

$p_i^{\min}$ Minimum generation of unit i.

Δ_i Maximum ramp-rate of unit i.

t_i^{up} Minimum uptime of unit i.

t_i^{down} Minimum downtime of unit i.

t_i^{cold} Cold startup time of unit i.

Decision Variables:

p_{it} Generation of unit i in time period t.

u_{it} 0-1 state variable of unit i in time period t ($u_{it} = 1$: on, $u_{it} = 0$: off).

v_{it} Number of time periods in which unit i has been on or off during time period t ($v_{it} > 0$: unit has been on, $v_{it} < 0$: unit has been off).

We denote the vectors $(u_{i1}, \ldots, u_{iT})^\top$ and $(p_{i1}, \ldots, p_{iT})^\top$ as $\boldsymbol{u_i}$ and $\boldsymbol{p_i}$, respectively. Moreover, $\boldsymbol{u}$ and $\boldsymbol{p}$ denote the schedules and generations of all units. Throughout the reminder of the present paper, one time period corresponds to one hour.

2.2 Formulation

The objective function of the UCP represents the total production cost over the planning horizon. The total production cost consists of the fuel cost and the startup cost.

The fuel cost of unit i in time period t is usually given as the following convex quadratic function of p_{it}:

$$f_i(p_{it}) = a_i p_{it}^2 + b_i p_{it} + c_i, \tag{2.1}$$

where coefficients a_i, b_i, and c_i are generally nonnegative.

The startup cost is related to the energy necessary to turn on a unit that has been off and occurs only when the unit is turned on during time period t ($u_{i,t-1} = 0$ and $u_{it} = 1$). In general, the startup cost depends on the number of the time periods that the unit has been off. In the present paper, we assume that two types of startup costs, S_i^{hot} and S_i^{cold}, are given. Here, S_i^{hot} represents the "hot startup cost", which is required when a hot off-unit is turned on (when the number $-v_{it}$ of time periods the unit has been off is below $t_i^{\text{down}} + t_i^{\text{cold}}$). On the other hand, S_i^{cold} represents the "cold startup cost", which is required when the cold off-unit is turned on (when the number of $-v_{it}$ of time periods the unit has been off is greater than $t_i^{\text{down}} + t_i^{\text{cold}}$). The startup cost of unit i in time period t is then given as

$$S_i(v_{i,t-1}, u_{it}, u_{i,t-1}) = \begin{cases} u_{it}(1 - u_{i,t-1})S_i^{\text{hot}} & \text{if } v_{i,t-1} < -t_i^{\text{down}} - t_i^{\text{cold}} \\ u_{it}(1 - u_{i,t-1})S_i^{\text{cold}} & \text{if } v_{i,t-1} \geq -t_i^{\text{down}} - t_i^{\text{cold}}. \end{cases} \tag{2.2}$$

Using (2.1) and (2.2), the objective function ϕ of the UCP is defined as

$$\phi(\boldsymbol{p}, \boldsymbol{u}) = \sum_{t=1}^{T} \sum_{i \in I} \{u_{it} f_i(p_{it}) + S_i(v_{i,t-1}, u_{it}, u_{i,t-1})\}. \tag{2.3}$$

The constraints of the UCP consist of the system constraints and operating constraints of individual units.

<u>System Constraints</u>

- Demand constraints:

$$D_t = \sum_{i \in I} u_{it} p_{it}, \ t = 1, \ldots, T. \tag{2.4}$$

- Spinning reserve constraints:

$$R_t \leq \sum_{i \in I} u_{it}(p_i^{\max} - p_{it}), \ t = 1, \ldots, T. \tag{2.5}$$

  The spinning reserve constraint is used in the case of an unexpected increase in the demand or a unit failure. Using (2.4), (2.5) is rewritten as

$$D_t + R_t \leq \sum_{i \in I} u_{it} p_i^{\max}, \ t = 1, \ldots, T. \tag{2.6}$$

<u>Operating Constraints of Thermal Unit</u>

- Generation limit constraints:

$$u_{it} p_i^{\min} \leq p_{it} \leq u_{it} p_i^{\max}, \ t = 1, \ldots, T. \tag{2.7a}$$

- Minimum uptime constraints:

$$u_{it} = 1 \ \text{ if } 1 \leq v_{i,t-1} < t_i^{\text{up}}, \ t = 1, \ldots, T. \tag{2.7b}$$

- Minimum downtime constraints:

$$u_{it} = 0 \ \text{ if } \ -1 \geq v_{i,t-1} > -t_i^{\text{down}}, \ t = 1, \ldots, T. \tag{2.7c}$$

- State transition equations for $t = 1, \ldots, T$:

$$v_{it} = \begin{cases} \min(t_i^{\text{up}}, \ \max(v_{i,t-1}, 0) + 1) & \text{if } u_{it} = 1 \\ \max(-t_i^{\text{down}} - t_i^{\text{cold}}, \ \min(v_{i,t-1}, 0) - 1) & \text{if } u_{it} = 0. \end{cases} \tag{2.7d}$$

- Ramp-rate limit constraints:

$$|p_{it} - u_{it} p_{i,t-1}| \leq \Delta_i, \ t = 1, \ldots, T. \tag{2.7e}$$

The minimum uptime (downtime) constraints mean that a unit must be on (off) for a certain number of time periods once it has been turned on (off). The ramp-rate limit constraints mean that a generation cannot change quickly.

Consequently, the UCP is formulated as

$$\mathbf{P}^0 : \ \begin{array}{ll} \min\limits_{\boldsymbol{p}, \boldsymbol{u}} & \phi(\boldsymbol{p}, \boldsymbol{u}) \\ \text{s.t.} & (2.4), (2.6), (2.7). \end{array}$$

The problem $\mathbf{P}^0$ is a mixed-integer programming problem including 0-1 variables. Here, $\mathbf{P}^0$ is in the class of NP-hard problems, and an exact optimal solution to this problem is difficult to obtain.

3 Lagrangian Relaxation Method

In this section, we explain the Lagrangian relaxation (LR) method for solving the UCP.

The LR method basically solves the dual problem of the UCP. First, we focus on the relationship between the primal and dual problem of the UCP. Let the dual problem of $\mathbf{P}^0$ be denoted by $\mathbf{D}^0$. Then, from the duality theory [3], the following inequality holds:

$$\min(\mathbf{P}^0) \geq \max(\mathbf{D}^0).$$

Hereinafter, we remove the ramp-rate limit constraints (2.7e) from $\mathbf{P}^0$ for simplicity. Let $\mathbf{P}$ denote $\mathbf{P}^0$ without (2.7e).

$$\mathbf{P} : \ \begin{array}{ll} \min\limits_{\boldsymbol{p}, \boldsymbol{u}} & \phi(\boldsymbol{p}, \boldsymbol{u}) \\ \text{s.t.} & D_t = \sum\limits_{i \in I} u_{it} p_{it}, \ t = 1, \ldots, T \\ & D_t + R_t \leq \sum\limits_{i \in I} u_{it} p_i^{\max}, \ t = 1, \ldots, T \\ & (\boldsymbol{p}_i, \boldsymbol{u}_i) \in U_i, \ \forall i \in I, \end{array}$$

where

$$U_i := \{(\boldsymbol{p}_i, \boldsymbol{u}_i) \mid (\boldsymbol{p}_i, \boldsymbol{u}_i) \text{ satisfies } (2.7a), (2.7b), (2.7c), and (2.7d)\}.$$

Since the feasible region of $\mathbf{P}$ includes that of $\mathbf{P}^0$, the following inequality holds:

$$\min(\mathbf{P}^0) \geq \min(\mathbf{P}).$$

Since the inequality

$$\min(\mathbf{P}) \geq \max(\mathbf{D})$$

holds by the duality theory, we have

$$\min(\mathbf{P}^0) \geq \max(\mathbf{D}).$$

Therefore, the optimal value of $\mathbf{D}$ provides the lower bound for $\mathbf{P}^0$ as well as $\mathbf{P}$. Note that if a feasible solution of $\mathbf{P}$ satisfies the ramp-rate limit constraints (2.7e), then it also is feasible for $\mathbf{P}^0$.

In the reminder of the present paper, we focus on the problem $\mathbf{P}$. Now, we define the Lagrangian function L as

$$L(\boldsymbol{p}, \boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := \phi(\boldsymbol{p}, \boldsymbol{u}) + \sum_{t=1}^{T} \lambda_t (D_t - \sum_{i \in I} u_{it} p_{it}) + \sum_{t=1}^{T} \mu_t (D_t + R_t - \sum_{i \in I} u_{it} p_i^{\max}),$$

where $\lambda_t \in \Re, t = 1, \ldots, T$ and $\mu_t \geq 0, t = 1, \ldots, T$ are the Lagrangian multipliers to (2.4) and (2.6), respectively, $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_T)^\top$ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_T)^\top$.

Using the Lagrangian function L, the dual problem of $\mathbf{P}$ can be written as

$$\begin{array}{lll} \mathbf{D}: & \max & \theta(\boldsymbol{\lambda}, \boldsymbol{\mu}) \\ & \text{s.t.} & \boldsymbol{\mu} \geq 0, \end{array}$$

where the dual function $\theta(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is defined by the optimal value of the Lagrangian relaxed problem:

$$\begin{array}{ll} \min\limits_{\boldsymbol{p}, \boldsymbol{u}} & L(\boldsymbol{p}, \boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \\ \text{s.t.} & (\boldsymbol{p}_i, \boldsymbol{u}_i) \in U_i, \ \forall i \in I, \end{array} \tag{3.1}$$

that is,

$$\theta(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\boldsymbol{p}, \boldsymbol{u}} \{L(\boldsymbol{p}, \boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \mid (\boldsymbol{p}_i, \boldsymbol{u}_i) \in U_i, \ i \in I\}.$$

If $\min(\mathbf{P}) > \max(\mathbf{D})$ holds, then the duality gap exists. In general, there exists a duality gap for the UCP. If $\phi(\boldsymbol{p}, \boldsymbol{u}) - \theta(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is small for given feasible solutions $(\boldsymbol{p}, \boldsymbol{u})$ and $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ of $\mathbf{P}$ and $\mathbf{D}$, respectively, then $(\boldsymbol{p}, \boldsymbol{u})$ is considered to be a good approximate solution of $\mathbf{P}$. Thus, we want $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ to be the maximum solution of $\mathbf{D}$.

Note that even if the maximum $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ of $\mathbf{D}$ is obtained, the solution $(\hat{\boldsymbol{p}}, \hat{\boldsymbol{u}})$ of the Lagrangian relaxed problem (3.1) with $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ is not always feasible for $\mathbf{P}$. Thus, we must find a feasible solution of $\mathbf{P}$ from the solution $(\hat{\boldsymbol{p}}, \hat{\boldsymbol{u}})$ of the problem (3.1). Therefore, the Lagrange relaxation method is executed in two steps: the maximization of the dual problem (Subsection 3.1) and the construction of a feasible solution (Subsection 3.2).

3.1 Maximization of the Dual Problem

The dual problem **D** is a nondifferentiable convex problem. In order to solve such a problem, the subgradient method [2] and the bundle method[6] are useful. In [6], it is reported that the bundle method requires fewer iterations to converge, as compared to the subgradient method. However, the quadratic problem must be solved in each iteration. Since the cost of solving the quadratic programming problem is much greater than that of solving the relaxed problem (3.1), the effect of the reduced number of iterations of the bundle method is limited. Therefore, in the present paper, we apply the standard subgradient method to solve the dual problem.

3.1.1 Subgradient Method for the Dual Problem

Here, we explain the subgradient method for the dual problem of the UCP. Now, let $(\boldsymbol{p}^k, \boldsymbol{u}^k)$ denote the solution of the Lagrangian relaxed problem (3.1) for $(\boldsymbol{\lambda}^k, \boldsymbol{\mu}^k)$. Then, the subgradient ξ^k of the dual function is given by

$$\xi^k = \begin{pmatrix} D_1 - \sum_{i\in I} p_{i1}^k \\ \vdots \\ D_T - \sum_{i\in I} p_{iT}^k \\ D_1 + R_1 - \sum_{i\in I} u_{i1}^k p_i^{\max} \\ \vdots \\ D_T + R_T - \sum_{i\in I} u_{iT}^k p_i^{\max} \end{pmatrix} \in \partial\theta(\boldsymbol{\lambda^k}, \boldsymbol{\mu^k}). \tag{3.2}$$

Using this subgradient ξ^k, the standard subgradient method updates Lagrangian multipliers as

$$\begin{pmatrix} \boldsymbol{\lambda}^{k+1} \\ \boldsymbol{\mu}^{k+1} \end{pmatrix} = \max\left\{0, \begin{pmatrix} \boldsymbol{\lambda}^{k} \\ \boldsymbol{\mu}^{k} \end{pmatrix} + \delta_t^k \frac{\xi^k}{\|\xi^k\|}\right\}, \tag{3.3}$$

where

$$\delta^k = \frac{1}{\epsilon + \sigma k}, \ \epsilon > 0, \sigma > 0. \tag{3.4}$$

Note that δ^k is the step size, and ϵ and σ are constant parameters.

The subgradient method for **D** is described as follows.

Subgradient Method for the Dual Problem D

Step 0: Select initial Lagrangian multipliers $(\boldsymbol{\lambda}^0, \boldsymbol{\mu}^0)$. Set $k := 0$.

Step 1: Solve problem (3.1) for $(\boldsymbol{\lambda}^k, \boldsymbol{\mu}^k)$ and obtain the solution $(\boldsymbol{p}^k, \boldsymbol{u}^k)$.

Step 2: If the stopping criteria is satisfied, then set $(\hat{\boldsymbol{p}}, \hat{\boldsymbol{u}}, \hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\mu}}) := (\boldsymbol{p}^k, \boldsymbol{u}^k, \boldsymbol{\lambda}^k, \boldsymbol{\mu}^k)$ and terminate. Otherwise, go to Step 3.

Step 3: Update $(\boldsymbol{\lambda}^{k+1}, \boldsymbol{\mu}^{k+1})$ by (3.3), set $k := k+1$, and go to Step 1.

3.1.2 Dynamic Programming for the Lagrangian Relaxed Problem

The main task of the subgradient method is to solve the Lagrangian relaxed problem (3.1). Next, we explain the solution of the Lagrangian relaxed problem (3.1) by dynamic programming (DP) .

First, note that the dual function is separated as

$$\theta(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{i \in I} \theta_i(\boldsymbol{\lambda}, \boldsymbol{\mu}) + \sum_{t=1}^{T} \{\lambda_t D_t + \mu_t (D_t + R_t)\},$$

where $\theta_i(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is defined by the optimal value of the following problem:

$$\begin{array}{ll} \min & \displaystyle\sum_{t=1}^{T} [u_{it}\{f_i(p_{it}) - \lambda_t p_{it} - \mu_t p_i^{\max}\} + S_i(v_{i,t-1}, u_{it}, u_{i,t-1})] \\ \text{s.t.} & (\boldsymbol{p}_i, \boldsymbol{u}_i) \in U_i. \end{array} \tag{3.5}$$

Problem (3.5) is a subproblem of unit i and can be solved efficiently by the following DP algorithm.

DP is an enumeration method based on the principle of optimality (see [4] for details on DP). The optimal solution of the problem (3.5) is obtained by enumerating the costs of all states from time period 1 to time period T. Recall that v_{it} denotes the state of unit i during time period t and that the number of all states during each time period is $t_i^{\text{down}} + t_i^{\text{cold}} + t_i^{\text{up}} + 1$. Let $C_i(t, v_i)$ denote the cost when the state of unit i during time period t is v_{it}. Now, assume that the costs $C_i(t-1, v_{i,t-1})$, $v_{i,t-1} = -t_i^{\text{down}} - t_i^{\text{cold}} - 1, \ldots, -1, 1, \ldots, t_i^{\text{up}}$ during time period $t-1$ are calculated. From (2.2), (2.7b), and (2.7c), the state of unit i must be one of the following:

1) Possible to turn on with cold startup cost S_i^{cold} in the next time period ($v_{i,t-1} = -t_i^{\text{down}} - t_i^{\text{cold}} - 1$),

2) Possible to turn on with hot startup cost S_i^{hot} in the next time period ($-t_i^{\text{down}} \geq v_{i,t-1} \geq -t_i^{\text{down}} - t_i^{\text{cold}}$),

3) Impossible to turn on in the next time period ($-1 \geq v_{i,t-1} > -t_i^{\text{down}}$),

4) Impossible to turn off in the next time period ($1 \leq v_{i,t-1} < t_i^{\text{up}}$),

5) Possible to turn off in the next time period ($v_{i,t-1} = t_i^{\text{up}}$).

Then, we can calculate the cost $C_i(t, v_{it})$ of the next time period t by the recursive formula:

$$C_i(t, v_{it}) = \begin{cases} F_i^0(t) + \min\{C_i(t-1, -t_i^{\text{down}} - t_i^{\text{cold}} - 1),\ C_i(t-1, -t_i^{\text{down}} - t_i^{\text{cold}})\} & \text{if } v_{it} = -t_i^{\text{down}} - t_i^{\text{cold}} - 1 \\ F_i^0(t) + C_i(t-1, v_{it}+1) & \text{if } -t_i^{\text{down}} - t_i^{\text{cold}} \leq v_{it} \leq -2 \\ F_i^0(t) + C_i(t-1, t_i^{\text{up}}) & \text{if } v_{it} = -1 \\ F_i^1(t) + \min\{C_i(t-1, -t_i^{\text{down}} - t_i^{\text{cold}} - 1) + S_i^{\text{cold}}, \\ \qquad C_i(t-1, -t_i^{\text{down}} - t_i^{\text{cold}}) + S_i^{\text{hot}}, \ldots, C_i(t-1, -t_i^{\text{down}}) + S_i^{\text{hot}}\} & \text{if } v_{it} = 1 \\ F_i^1(t) + C_i(t-1, v_{it}-1) & \text{if } 2 \leq v_{it} \leq t_i^{\text{up}} - 1 \\ F_i^1(t) + \min\{C_i(t-1, t_i^{\text{up}}),\ C_i(t, t_i^{\text{up}} - 1)\} & \text{if } v_{it} = t_i^{\text{up}}, \end{cases} \tag{3.6}$$

where $F_i^0(t)$ is the cost when unit i is off during time period t ($F_i^0(t) = 0$ in general), and $F_i^1(t)$ is the fuel cost when unit i is on during time period t. The fuel cost $F_i^1(t)$ is given as the optimal value of the convex quadratic programming problem:

$$\begin{array}{ll} \min & a_i p_{it}^2 + b_i p_{it} + c_i - \lambda_t p_{it} - \mu_t p_i^{\max} \\ \text{s.t.} & p_i^{\min} \leq p_{it} \leq p_i^{\max}. \end{array} \tag{3.7}$$

Therefore,

$$F_i(t) = a_i\tilde{p}_{it}^2 + b_i\tilde{p}_{it} + c_i - \lambda_t\tilde{p}_{it} - \mu_t p_i^{\max},$$

where

$$\tilde{p}_{it} = \max\left\{p_i^{\min},\ \min\left\{p_i^{\max},\ \frac{-b_i+\lambda_t}{2a_i}\right\}\right\}.$$

Figure 1 shows part of the transition graph of unit i having the minimum uptime of four time periods, a minimum downtime of three time periods, and a cold startup time of two time periods.

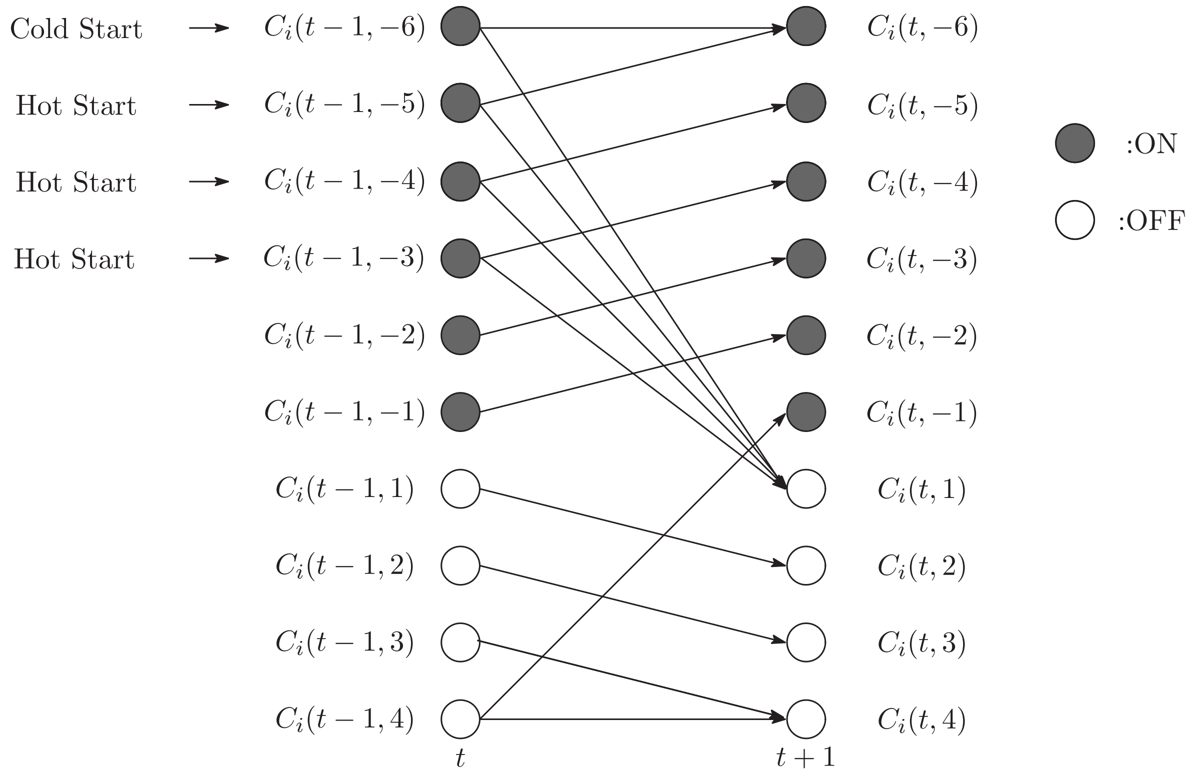

Figure 1: State transition graph for two time periods in DP ($t_i^{\rm up} = 4, t_i^{\rm down} = 3, t_i^{\rm cold} = 2$)

When an initial state $\bar{v}_i$ is given, we set $C_i(0, v_{i0})$ as

$$C_i(0, v_{i0}) = \begin{cases} 0 & \text{if } v_{i0} = \bar{v}_i \\ \infty & \text{if } v_{i0} \neq \bar{v}_i. \end{cases} \tag{3.8}$$

In the first step of DP, we calculate $C_i(t, v_{it})$, $v_{it} = -t_i^{\rm down} - t_i^{\rm cold} - 1, \dots, -1, 1, \dots, t_i^{\rm up}$ from time period 1 to time period T using the recursive formula (3.6) and save the state transitions during every time period. Next, we choose the final state v_{iT}^* as the minimizer of $C_i(T, \cdot)$. The minimum value of problem (3.5) is then $C_i(T, v_{iT}^*)$. Finally, we trace the state transitions backward from v_{iT}^* and obtain the optimal schedule $\boldsymbol{u_i}$ and generation $\boldsymbol{p_i}$. The computation cost of the subproblem (3.5) is $O(T \times (t_i^{\rm up} + t_i^{\rm down} + t_i^{\rm cold}))$ if we simply calculate the all costs $C_i(t, v_{it})$, $t = 1, \dots, T$, $v_{it} = -t_i^{\rm down} - t_i^{\rm cold} - 1, \dots, -1, 1, \dots, t_i^{\rm up}$. The cost can be reduced to $O(T \times t_i^{\rm cold})$ so that it is only necessary to calculate the costs in state $v_{it} = -t_i^{\rm down} - t_i^{\rm cold} - 1, \dots, -t_i^{\rm down}, t_i^{\rm up}$.

3.2 Construction of a Feasible Solution

Let $(\hat{\boldsymbol{p}}, \hat{\boldsymbol{u}})$ be the solution of the Lagrangian relaxed problem (3.1) for $(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\mu}})$. If $(\hat{\boldsymbol{p}}, \hat{\boldsymbol{u}})$ satisfies the demand constraints (2.4) and the spinning reserve constraints (2.6), then $(\hat{\boldsymbol{p}}, \hat{\boldsymbol{u}})$ is a feasible solution of the primal problem **P**. However, in general, $(\hat{\boldsymbol{p}}, \hat{\boldsymbol{u}})$ does not satisfy (2.4) and (2.6), and we have to find a feasible solution of **P** from $(\hat{\boldsymbol{p}}, \hat{\boldsymbol{u}})$.

A schedule that satisfies the spinning reserve constraints (2.6) is called "reserve-feasible". Next, we assume that the reserve-feasible schedule $\boldsymbol{u}^*$ is created from $\hat{\boldsymbol{u}}$. Then, $\boldsymbol{u}^*$ satisfies

$$D_t + R_t \leq \sum_{i \in I} u_{it}^* p_i^{\max}, \ t = 1, \dots, T.$$

Moreover, if $(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\mu}})$ is a good approximate solution to the dual problem, then the reserve-feasible schedule $\boldsymbol{u}^*$ generally satisfies the following equation:

$$\sum_{i \in I} u_{it}^* p_i^{\min} \leq D_t, \ t = 1, \dots, T.$$

Therefore, there exist the generation $\boldsymbol{p}$ satisfying the demand constraints (2.4).

The optimal generation for the schedule $\boldsymbol{u}^*$ is obtained by solving the economic dispatching (ED) problem:

$$\begin{aligned} \textbf{ED}: \ \min_{\boldsymbol{p}} \quad & \sum_{t=1}^{T} \sum_{i \in I} u_{it}^* f_i(p_{it}) \\ \text{s.t.} \quad & D_t = \sum_{i \in I} u_{it}^* p_{it}, \ t = 1, \dots, T \\ & u_{it}^* p_i^{\min} \leq p_{it} \leq u_{it}^* p_i^{\max}, \ \forall i \in I, \ t = 1, \dots, T. \end{aligned}$$

The problem **ED** is **P** with the schedule $\boldsymbol{u}^*$ fixed and is a convex quadratic programming problem. The startup costs in the objective function and the minimum up/down constraints are removed because they are constant or are satisfied. **ED** can be solved efficiently by using its special structure. The technique to solve **ED** is given in Subsection 3.2.2.

A feasible solution $(\boldsymbol{p}^*, \boldsymbol{u}^*)$ is obtained through the following steps.

Construction of a Feasible Solution

Step 1. Find a reserve-feasible schedule $\boldsymbol{u}^*$ from $\hat{\boldsymbol{u}}$ ($(\hat{\boldsymbol{p}}, \hat{\boldsymbol{u}})$ is the solution of the Lagrangian relaxed problem for $(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\mu}})$).

Step 2. Obtain a feasible generation $\boldsymbol{p}^*$ by solving **ED** with schedule $\boldsymbol{u}^*$.

3.2.1 Finding a Reserve-Feasible Schedule

For a given schedule $\hat{\boldsymbol{u}}$, we consider the following function:

$$R^{\text{def}}(t, \hat{\boldsymbol{u}}) = D_t + R_t - \sum_{i \in I} \hat{u}_{it} p_i^{\max}.$$

$R^{\text{def}}(t, \hat{\boldsymbol{u}})$ represents the amount of the reserve deficit in time period t. If $R^{\text{def}}(t, \hat{\boldsymbol{u}}) \leq 0$, then $\hat{\boldsymbol{u}}$ is reserve-feasible in time period t. On the other hand, if $R^{\text{def}}(t, \hat{\boldsymbol{u}}) > 0$, then $\hat{\boldsymbol{u}}$ is not reserve-feasible in time period t. Moreover, the larger the value of $R^{\rm def}(t, \hat{\boldsymbol{u}})$, the greater the reserve deficit. Thus, we must make a schedule $\boldsymbol{u}^*$ that satisfies $R^{\rm def}(t, \boldsymbol{u}^*) \leq 0$ in every time period.

Several methods for finding a reserve-feasible schedule from the solution of the Lagrangian relaxed problem have been proposed. For example, [17] proposed increasing the Lagrangian multiplier μ_t corresponding to the time period t with the maximum reserve deficit and repeating this until the reserve-feasible schedule is obtained. [17] also presented a method by which to calculate the exact amount of the increase of μ_t needed to turn on a certain unit. However, even if we can increase μ_t exactly, the units that have the same properties are turned on simultaneously. Moreover, there is another disadvantage. It is difficult to determine the exact amount of the increase of μ_t if there exists another schedule with the same total production cost for certain Lagrangian multipliers $\boldsymbol{\mu}$.

In the present paper, we propose local search methods for improving the obtained feasible solution. Therefore, it is not necessary to obtain the "best" reserve-feasible schedule by a complicated algorithm. As such, we use the following simple method. The basic idea of this method is to turn on a unit that is off in the current schedule $\hat{u}$ compulsively. Let $t^{\rm def}$ be the time period in which the reserve deficit is the maximum. We turn on an off-unit i (that is, $\hat{u}_{it^{\rm def}} = 0$) and repeat it until the reserve-feasible schedule is obtained. However, just turning on an off-unit might violate the minimum uptime and downtime constraints (2.7b) and (2.7c), respectively. To avoid such violations, we consider the problem of unit i:

$$\begin{array}{ll} \min\limits_{\boldsymbol{p_i},\boldsymbol{u_i}} & \sum\limits_{t=1}^{T} [u_{it}\{f_i(p_{it}) - \hat{\lambda}_t p_{it} - \hat{\mu}_t p_i^{\max}\} + S_i(v_{i,t-1}, u_{it}, u_{i,t-1})] \\ \text{s.t.} & (\boldsymbol{p}_i, \boldsymbol{u}_i) \in U_i, \ \forall i \in I \\ & u_{it^{\rm def}} = 1 \\ & u_{it} \geq \hat{u}_{it}, \ t = 1, \ldots, T. \end{array} \tag{3.9}$$

In problem (3.9), the second constraint means that unit i must be turned on during time period $t^{\rm def}$ ($u_{it^{\rm def}}$ is set to 1 compulsively), and the third constraint means that unit i must be on if it is on in the current schedule $\hat{\boldsymbol{u}}_i$. Since problem (3.9) is similar to problem (3.5), we can solve this problem using DP, as explained in Subsection 3.1.2. For the time period t, when unit i must be on, we set $F_i^0(t)$ in (3.6) to ∞ in order to satisfy the second and third constraints of the problem (3.9), that is, we set

$$F_i^0(t) := \begin{cases} \infty & \text{if } t = t^{\rm def} \text{ or } \hat{u}_{it} = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Then, we can solve problem (3.9) in a similar manner as problem (3.5).

Let $(\boldsymbol{p}'_i, \boldsymbol{u}'_i)$ and $\theta'(i)$ denote the optimal solution and optimal value of the problem (3.9), respectively, and let $\boldsymbol{u}'$ denote the schedule obtained by replacing $\hat{\boldsymbol{u}}_i$ with $\boldsymbol{u}'_i$. Then, the schedule $\boldsymbol{u}'$ satisfies $R^{\rm def}(t, \boldsymbol{u}') \leq R^{\rm def}(t, \hat{\boldsymbol{u}})$, $t = 1, \ldots, T$. Moreover, $R^{\rm def}(t^{\rm def}, \boldsymbol{u}') < R^{\rm def}(t^{\rm def}, \hat{\boldsymbol{u}})$ always holds.

The method to find a reserve-feasible schedule used in the present paper is as follows.

Finding a Reserve-Feasible Schedule

Step 0: Obtain an approximate solution $\hat{\boldsymbol{u}}$ by solving the dual problem **D**.

Step 1: Calculate $R^{\text{def}}(t, \hat{\boldsymbol{u}})$, $t = 1, \dots, T$. If $R^{\text{def}}(t, \hat{\boldsymbol{u}}) \leq 0$ holds for all $t = 1, \dots, T$, then $\boldsymbol{u}^* := \hat{\boldsymbol{u}}$ and terminate.

Step 2: Choose the time period t^{def}, for which the spinning reserve constraint is most violated.

$$t^{\text{def}} := \arg\max\{R^{\text{def}}(t, \hat{\boldsymbol{u}}),\ t = 1, \dots, T\} \tag{3.10}$$

Define the set of units that are off in time period t^{def}.

$$I^{\text{def}} := \{i \in I \mid \hat{u}_{it^{\text{def}}} = 0\} \tag{3.11}$$

Step 3: For all $i \in I^{\text{def}}$, obtain the solution $(\boldsymbol{p}'_i, \boldsymbol{u}'_i)$ of the problem (3.9) and its optimal value $\theta'(i)$.

Step 4: Choose the unit with the smallest value of $\theta'(i)$.

$$j := \arg\min\{\theta'(i),\ i \in I^{\text{def}}\} \tag{3.12}$$

Step 5: Set $\hat{\boldsymbol{u}}_j := \boldsymbol{u}'_j$, and go to Step 1.

3.2.2 Economic Dispatch

In this subsection, we explain an efficient technique to solve the economic dispatching problem **ED**.

Problem **ED** is decomposed into the following subproblem for a time period t.

$$\begin{aligned} \mathbf{ED}(t): \quad & \min \quad \sum_{i \in I} u^*_{it} f_i(p_{it}) \\ & \text{s.t.} \quad D_t = \sum_{i \in I} u^*_{it} p_{it} \\ & \qquad\ \ u^*_{it} p_i^{\min} \leq p_{it} \leq u^*_{it} p_i^{\max},\ \forall i \in I, \end{aligned} \tag{3.13}$$

where $\boldsymbol{u}^*$ is a given schedule. Since the objective function of $\mathbf{ED}(t)$ is convex and the constraints are linear, there is no duality gap.

The dual problem of $\mathbf{ED}(t)$ can be formulated as

$$\max_{\lambda_t \in \Re} \quad \psi(\lambda_t), \tag{3.14}$$

where $\psi(\lambda_t)$ is defined by the optimal value of the problem:

$$\begin{aligned} \min \quad & \sum_{i \in I} u^*_{it}(a_i p_{it}^2 + b_i p_{it} + c_i) + \lambda_t (D_t - \sum_{i \in I} u^*_{it} p_{it}) \\ \text{s.t.} \quad & u^*_{it} p_i^{\min} \leq p_{it} \leq u^*_{it} p_i^{\max},\ \forall i \in I. \end{aligned} \tag{3.15}$$

Problem (3.15) is decomposable by individual units, and its optimal solution is given by

$$\tilde{p}_{it} = \begin{cases} 0 & \text{if } u^*_{it} = 0 \\ \max\left\{p_i^{\min},\ \min\left\{p_i^{\max},\ \frac{-b_i + \lambda_t}{2a_i}\right\}\right\} & \text{if } u^*_{it} = 1, \end{cases} \tag{3.16}$$

for all $i \in I$. Since the problem (3.14) is an unconstrained optimization problem with only one variable, we can solve it efficiently using the quasi-Newton method (see [3, Section 8.6] for the quasi-Newton method). In the quasi-Newton method, the gradient of the objective function $\nabla\psi(\lambda_t^k)$ is necessary. The objective function of the dual problem is generally not differentiable, However, if problem (3.15) has a unique solution for an arbitrary Lagrangian multiplier, then the objective function ψ is continuously differentiable. Problem (3.15) satisfies this condition because it is a strict convex quadratic programming problem. The gradient of ψ is given by

$$\nabla\psi(\lambda_t^k) = D_t - \sum_{i\in I} \tilde{p}_{it}. \tag{3.17}$$

The optimal solution λ_t^* of (3.14) satisfies $\nabla\psi(\lambda_t^*) = 0$. From (3.17), the solution $p_{it}^*,\ i \in I$ of the problem (3.15) for λ_t^* satisfies the demand constraint (2.4). Therefore, $p_{it}^*,\ i \in I$ is the solution of $\mathbf{ED}(t)$.

The quasi-Newton method to solve the dual problem (3.14) is described as follows.

Quasi-Newton Method for the Dual Problem of ED(t)

Step 0: Choose an initial point λ_t^0 and H^0. Set $k := 0$.

Step 1: If λ_t^k is the optimal solution, then terminate. Otherwise, go to Step 2.

Step 2: Calculate $\nabla\psi(\lambda_t^k)$ using (3.16) and (3.17). Set $d^k := -H^k\nabla\psi(\lambda_t^k)$.

Step 3: Determine the stepsize s_k and set $\lambda_t^{k+1} := \lambda_t^k + s_k d^k$.

Step 4: Update $H^{k+1} := H^k$ using the BFGS formula. Set $k := k + 1$ and go to Step 1.

Note that H^k denotes the inverse of the approximate matrix of the Hessian of the objective function. Moreover, H^k is scalar because the dimension of problem (3.14) is 1. The quasi-Newton method converges very quickly if the initial point λ_t^0 and H^0 are chosen correctly. The optimal solution λ_t^* of problem (3.14) is expected to be close to $\hat{\lambda}_t$, which is the solution of problem (3.1). Therefore, we set the initial point as $\lambda_t^0 := \hat{\lambda}_t$ in the numerical experiments.

4 New Local Search Methods for the Unit Commitment Problem

The feasible solution $(\boldsymbol{p}^*, \boldsymbol{u}^*)$ obtained by the LR method is often unsatisfactory compared to the exact optimal solution. As for the reason why the LR-based solution is not good, the following possibilities are considered:

- In the step of maximization of the dual problem (Subsection 3.1), relaxed problem (3.1) is decomposed into subproblems, which are solved independently. Therefore, the same schedules are obtained for units with the same properties.
- In the step of finding a reserve-feasible schedule (Subsection 3.2), some off-units are turned on not only during the time period with a reserve deficit but also before or after that time period due to the minimum up/down constraints.

There exist excessive on-units in the schedule obtained by the LR method because of the above reasons. Thus, the solution would be improved by changing the schedule of individual units. Therefore, we propose a local search method for improving the LR-based feasible solution $(\boldsymbol{p}^*, \boldsymbol{u}^*)$.

For a given feasible solution z, the local search method (LS) is to find a better solution z' in the neighborhood $N(z)$ of z and replace z with z'. This process is repeated until there exists no better solution in the neighborhood. For the UCP, the optimal generation is uniquely determined by solving the convex problem **ED** if the schedule is determined. Therefore, in the present, we consider a local search changing an individual schedule from the LR-based feasible solution. We propose two types of neighborhoods: the one-unit neighborhood and the two-unit neighborhood. The one-unit neighborhood is defined as the feasible set in which the schedule of all units except one are fixed, and the two-unit neighborhood is defined as the feasible set in which the schedules of all units except two are fixed. We herein refer to the proposed local search methods using these neighborhoods as the one-unit local search and the two-unit local search, respectively.

4.1 One-unit Local Search

In this subsection, we propose the one-unit local search (LS1).

First, we define the set Ω as

$$\Omega := \{(\boldsymbol{p}, \boldsymbol{u}) \mid (\boldsymbol{p}_i, \boldsymbol{u}_i) \in U_i, i \in I,\ (\boldsymbol{p}, \boldsymbol{u}) \text{ satisfies (2.4) and (2.6)}\}.$$

The set Ω is the feasible set of the problem **P**. The proposed method restricts the neighborhood search only in Ω.

For a given feasible solution $(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$, we define the feasible set $N_1^{\tilde{i}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ as

$$N_1^{\tilde{i}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}}) := \{(\boldsymbol{p}, \boldsymbol{u}) \in \Omega \mid \boldsymbol{u}_i = \bar{\boldsymbol{u}}_i, i \in I \setminus \{\tilde{i}\}\}.$$

The schedules of all units except unit $\tilde{i}$ are fixed to $\bar{\boldsymbol{u}}$ in the set $N_1^{\tilde{i}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$. We define one-unit neighborhood $N_1(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ as the union of sets $N_1^{\tilde{i}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}}), \tilde{i} \in I$, that is,

$$N_1(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}}) := \bigcup_{\tilde{i} \in I} N_1^{\tilde{i}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}}).$$

The candidate solutions in one-unit neighborhood $N_1(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ are obtained by enumerating the minimum cost solutions in $N_1^{\tilde{i}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ for all $\tilde{i} \in I$. The minimum cost solution in $N_1^{\tilde{i}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ for specified $\tilde{i}$ is a solution of the following problem:

$$\begin{aligned}
\mathbf{P}(\tilde{i}): \quad \min_{\boldsymbol{p}, \boldsymbol{u}_{\tilde{i}}} \quad & \sum_{t=1}^{T} \sum_{i \in I \setminus \{\tilde{i}\}} \bar{u}_{it} f_i(p_{it}) + \sum_{t=1}^{T} \{u_{\tilde{i}t} f_k(p_{\tilde{i}t}) + S_{\tilde{i}}(v_{\tilde{i},t-1}, u_{\tilde{i}t}, u_{\tilde{i},t-1})\} \\
\text{s.t.} \quad & D_t = \sum_{i \in I} \bar{u}_{it} p_{it} + u_{\tilde{i}t} p_{\tilde{i}t},\ t = 1, \ldots, T \\
& D_t + R_t \leq \sum_{i \in I \setminus \{\tilde{i}\}} \bar{u}_{it} p_i^{\max} + u_{\tilde{i}t} p_{\tilde{i}}^{\max},\ t = 1, \ldots, T \\
& \bar{u}_{it} p_i^{\min} \leq p_{it} \leq \bar{u}_{it} p_i^{\max},\ \forall i \in I \setminus \{\tilde{i}\},\ t = 1, \ldots, T \\
& (\boldsymbol{p}_{\tilde{i}}, \boldsymbol{u}_{\tilde{i}}) \in U_{\tilde{i}}.
\end{aligned}$$

Problem $\mathbf{P}(\tilde{i})$ is deduced from $\mathbf{P}$ by fixing the schedules of all units except unit $\tilde{i}$. The startup costs of all units except unit $\tilde{i}$ are removed because they are constant. $\mathbf{P}(\tilde{i})$ is considered as a small unit commitment problem for unit $\tilde{i}$ and is solved using DP. DP in Subsection 3.1.2 can be applied if $F^1_{\tilde{i}}(t)$ and $F^0_{\tilde{i}}(t)$ in (3.6), which are the costs when unit $\tilde{i}$ is on and off, respectively, during time period t, are specified for $\mathbf{P}(\tilde{i})$. We define the function $g_{\tilde{i}}(t; u_{\tilde{i}t})$ for given $u_{\tilde{i}t}$ by the optimal value of the problem:

$$\begin{array}{ll}\min & \displaystyle\sum_{i \in I \setminus \{\tilde{i}\}} \bar{u}_{it} f_i(p_{it}) + u_{\tilde{i}t} f_{\tilde{i}}(p_{\tilde{i}t}) \\ \text{s.t.} & D_t = \displaystyle\sum_{i \in I \setminus \{\tilde{i}\}} \bar{u}_{it} p_{it} + u_{\tilde{i}t} p_{\tilde{i}t} \\ & u_{\tilde{i}t} p^{\min}_{\tilde{i}} \le p_{\tilde{i}t} \le u_{\tilde{i}t} p^{\max}_{\tilde{i}} \\ & \bar{u}_{it} p^{\min}_i \le p_{it} \le \bar{u}_{it} p^{\max}_i, \ \forall i \in I \setminus \{\tilde{i}\}. \end{array} \tag{4.1}$$

Taking into account the feasibility of the problem $\mathbf{P}(\tilde{i})$, $F^1_{\tilde{i}}(t)$ and $F^0_{\tilde{i}}(t)$ are given by

$$F^1_{\tilde{i}}(t) := g_{\tilde{i}}(t; 1),$$
$$F^0_{\tilde{i}}(t) := \begin{cases} \infty & \text{if } D_t + R_t > \displaystyle\sum_{i \in I \setminus \{\tilde{i}\}} \bar{u}_{it} p^{\max}_i \\ g_{\tilde{i}}(t; 0) & \text{otherwise.} \end{cases}$$

If turning off unit $\tilde{i}$ violates of the spinning reserve constraint, then $F^0_{\tilde{i}t} = \infty$. Thus, the second constraint of the problem $\mathbf{P}(\tilde{i})$ is achieved.

For the solution $(\boldsymbol{p}', \boldsymbol{u}'_{\tilde{i}})$ of the problem $\mathbf{P}(\tilde{i})$, we set

$$\boldsymbol{p}^{\tilde{i}}_i := \boldsymbol{p}', \quad \boldsymbol{u}^{\tilde{i}}_i := \begin{cases} \bar{\boldsymbol{u}}_i & \text{if } i \in I \setminus \{\tilde{i}\} \\ \boldsymbol{u}'_i & \text{if } i = \tilde{i}. \end{cases}$$

If $\phi(\boldsymbol{p}^{\tilde{i}}, \boldsymbol{u}^{\tilde{i}}) < \phi(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$, then $(\boldsymbol{p}^{\tilde{i}}, \boldsymbol{u}^{\tilde{i}})$ is a better solution.

The optimization problem (4.1) must be solved in order to calculate $g_{\tilde{i}}(t; u_{\tilde{i}t})$. Problem (4.1) can be solved exactly using the same technique for $\mathbf{ED}(t)$. In practice, problem $\mathbf{P}(\tilde{i})$ has to be solved several times in LS1. Then, the costs $F^1_{\tilde{i}}(t)$ and $F^0_{\tilde{i}}(t)$ must be obtained rapidly. Thus, calculating $g_{\tilde{i}}(t; u_{\tilde{i}t})$ approximately would be better for large-scale problems (see Appendix A.1). However, the generation $\boldsymbol{p}'$ might violate the constraints if $g_{\tilde{i}}(t; u_{\tilde{i}t})$ is calculated approximately. Therefore, we solve problem $\mathbf{ED}$ with $\boldsymbol{u}^{\tilde{i}}$ in order to obtain the feasible generation $\boldsymbol{p}^{\tilde{i}}$ if $\phi(\boldsymbol{p}', \boldsymbol{u}'_{\tilde{i}}) < \phi(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$.

There could exist several better solutions in the neighborhood $N_1(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$. Therefore, we consider the following two typical strategies:

1. First admissible move strategy: search in the neighborhood $N^{\tilde{i}}_1(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ by some order of I and move to the first found better solution.

2. Best admissible move strategy: find all better solutions in the neighborhood $N_1(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ and move to the best solution.

We refer to the one-unit local search based on the first admissible move strategy as LS1-first and the one-unit local search based on the best admissible move strategy as LS1-best. The algorithms of LS1-first and LS1-best are as follows.

One-unit Local Search Based on the First Move Admissible Strategy (LS1-first)

Step 0: Let an initial feasible solution $(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ be given.

Step 1: Set $I^0 := I$, and choose a unit $\tilde{i} \in I^0$.

Step 2: Solve problem $\mathbf{P}(\tilde{i})$ and obtain the candidate solution $(\boldsymbol{p}^{\tilde{i}}, \boldsymbol{u}^{\tilde{i}})$.

Step 3: If $\phi(\boldsymbol{p}^{\tilde{i}}, \boldsymbol{u}^{\tilde{i}}) < \phi(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$, then set $(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}}) := (\boldsymbol{p}^{\tilde{i}}, \boldsymbol{u}^{\tilde{i}})$ and go to Step1.

Step 4: Set $I^0 := I^0/\{\tilde{i}\}$. If $I^0 = \emptyset$, then terminate. Otherwise, choose another unit $\tilde{i} \in I^0$ and go to Step 2.

One-unit Local Search Based on the Best Admissible Move Strategy (LS1-best)

Step 0: Let an initial feasible solution $(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ be given.

Step 1: Set $I^0 := I$, and choose an unit $\tilde{i} \in I^0$.

Step 2: Solve problem $\mathbf{P}(\tilde{i})$ and obtain the candidate solution $(\boldsymbol{p}^{\tilde{i}}, \boldsymbol{u}^{\tilde{i}})$.

Step 3: Set $I^0 := I^0/\{\tilde{i}\}$. If $I^0 = \emptyset$, then go to Step 4. Otherwise, choose another unit $\tilde{i} \in I^0$ and go to Step 2.

Step 4: Choose the most improved solution.

$$(\boldsymbol{p}^{\tilde{i}}, \boldsymbol{u}^{\tilde{i}}) := \arg\min\{\phi(\boldsymbol{p}^{\tilde{i}}, \boldsymbol{u}^{\tilde{i}}) \mid \tilde{i} \in I\}$$

Step 5: If $\phi(\boldsymbol{p}^{\tilde{i}}, \boldsymbol{u}^{\tilde{i}}) < \phi(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$, then $(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}}) := (\boldsymbol{p}^{\tilde{i}}, \boldsymbol{u}^{\tilde{i}})$ and go to Step 1. Otherwise, terminate.

4.2 Two-unit Local Search

In this subsection, we propose two-unit local search (LS2). LS2 is a natural extension of LS1.

First, we define the set I^2 as

$$I^2 := \{(i, j) \in I \times I \mid i < j\}$$

I^2 is the set of the combinations of two units. For a given feasible solution $(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$, we define the feasible set $N_2^{\tilde{i}\tilde{j}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ as

$$N_2^{\tilde{i}\tilde{j}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}}) := \{(\boldsymbol{p}, \boldsymbol{u}) \in \Omega \mid \boldsymbol{u}_i = \bar{\boldsymbol{u}}_i, i \in I \setminus \{\tilde{i}, \tilde{j}\}\}.$$

The schedules of all units except units $\tilde{i}$ and $\tilde{j}$ are fixed to $\bar{\boldsymbol{u}}$ in the set $N_2^{\tilde{i}\tilde{j}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$. We define two-unit neighborhood $N_2(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ as the union of sets $N_2^{\tilde{i}\tilde{j}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}}), (\tilde{i}, \tilde{j}) \in I^2$, that is,

$$N_2(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}}) := \bigcup_{(\tilde{i}, \tilde{j}) \in I^2} N_2^{\tilde{i}\tilde{j}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$$

Similar to the case of LS1, the candidate solutions in two-unit neighborhood $N_2(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ are obtained by enumerating the minimum solutions in $N_2^{\tilde{i}\tilde{j}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ for all $(\tilde{i}, \tilde{j}) \in I^2$. The minimum solution in

$N_2^{\tilde{i}\tilde{j}}(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ for specified $(\tilde{i}, \tilde{j})$ is obtained by solving the following problem:

$$\begin{aligned}
\mathbf{P}(\tilde{i},\tilde{j}): \quad \min_{\boldsymbol{p},\boldsymbol{u}_{\tilde{i}},\boldsymbol{u}_{\tilde{j}}} \quad & \sum_{t=1}^{T} \sum_{i\in I\setminus\{\tilde{i},\tilde{j}\}} \bar{u}_{it} f_i(p_{it}) + \sum_{t=1}^{T} \sum_{i\in\{\tilde{i},\tilde{j}\}} \{u_{it} f_i(p_{it}) + S_i(v_{i,t-1}, u_{it}, u_{i,t-1})\} \\
\text{s.t.} \quad & D_t = \sum_{i\in I\setminus\{\tilde{i},\tilde{j}\}} \bar{u}_{it} p_{it} + \sum_{i\in\{\tilde{i},\tilde{j}\}} u_{it} p_{it}, \ t = 1, \ldots, T \\
& D_t + R_t \le \sum_{i\in I\setminus\{\tilde{i},\tilde{j}\}} \bar{u}_{it} p_i^{\max} + \sum_{i\in\{\tilde{i},\tilde{j}\}} u_{it} p_i^{\max}, \ t = 1, \ldots, T \\
& \bar{u}_{it} p_i^{\min} \le p_{it} \le \bar{u}_{it} p_i^{\max}, \ \forall i \in I \setminus \{\tilde{i},\tilde{j}\}, \ t = 1, \ldots, T \\
& (\boldsymbol{p}_i, \boldsymbol{u}_i) \in U_i, \ \forall i \in \{\tilde{i},\tilde{j}\}
\end{aligned}$$

The problem $\mathbf{P}(\tilde{i}, \tilde{j})$ is deduced from $\mathbf{P}$ by fixing the schedules of all units except units $\tilde{i}$ and $\tilde{j}$ and is regarded as a natural extension of $\mathbf{P}(\tilde{i})$ in LS1. Since $\mathbf{P}(\tilde{i}, \tilde{j})$ is a unit commitment problem with two units and has twice the number of 0-1 variables of $\mathbf{P}(\tilde{i})$, we cannot use directly the same DP for $\mathbf{P}(\tilde{i})$. However, DP in Subsection 3.1.2 can be extended to the case of two units. The costs for the on/off states of units $\tilde{i}$ and $\tilde{j}$ in time period t should be calculated for the extended DP. We define the functions $g_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t}, u_{\tilde{j}t})$ for $u_{\tilde{i}t}$ and $u_{\tilde{j}t}$ based on the optimal value of the problem:

$$\begin{aligned}
\min \quad & \sum_{i\in I\setminus\{\tilde{i},\tilde{j}\}} \bar{u}_{it} f_i(p_{it}) + \sum_{i\in\{\tilde{i},\tilde{j}\}} u_{it} f_i(p_{it}) \\
\text{s. t.} \quad & D_t = \sum_{i\in I\setminus\{\tilde{i},\tilde{j}\}} \bar{u}_{it} p_{it} + \sum_{i\in\{\tilde{i},\tilde{j}\}} u_{it} p_{it} \\
& u_{it} p_i^{\min} \le p_{it} \le u_{it} p_i^{\max}, \ \forall i \in \{\tilde{i},\tilde{j}\} \\
& \bar{u}_{it} p_i^{\min} \le p_{it} \le \bar{u}_{it} p_i^{\max}, \ \forall i \in I \setminus \{\tilde{i},\tilde{j}\}.
\end{aligned} \tag{4.2}$$

Note that $g_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t}, u_{\tilde{j}t})$ is a natural extension of $g_{\tilde{i}}(t; u_{\tilde{i}t})$. Let $F_{\tilde{i}\tilde{j}}^{u_{\tilde{i}t}u_{\tilde{j}t}}(t)$ denote the cost in time period t when the states of unit $\tilde{i}$ and $\tilde{j}$ are $u_{\tilde{i}t}$ and $u_{\tilde{j}t}$, respectively. Then, taking the feasibility of problem $\mathbf{P}(\tilde{i}, \tilde{j})$ into account, we have

$$\begin{aligned}
F_{\tilde{i}\tilde{j}}^{11}(t) &:= g_{\tilde{i}\tilde{j}}^{2}(t; 1, 1), \\
F_{\tilde{i}\tilde{j}}^{10}(t) &:= \begin{cases} \infty & \text{if } D_t + R_t > \sum_{i\in I\setminus\{\tilde{i},\tilde{j}\}} \bar{u}_{it} + p_{\tilde{i}}^{\max} \\ g_{\tilde{i}\tilde{j}}(t; 1, 0) & \text{otherwise} \end{cases} \\
F_{\tilde{i}\tilde{j}}^{01}(t) &:= \begin{cases} \infty & \text{if } D_t + R_t > \sum_{i\in I\setminus\{\tilde{i},\tilde{j}\}} \bar{u}_{it} + p_{\tilde{j}}^{\max} \\ g_{\tilde{i}\tilde{j}}(t; 0, 1) & \text{otherwise} \end{cases} \\
F_{\tilde{i}\tilde{j}}^{00}(t) &:= \begin{cases} \infty & \text{if } D_t + R_t > \sum_{i\in I\setminus\{\tilde{i},\tilde{j}\}} \bar{u}_{it} \\ g_{\tilde{i}\tilde{j}}(t; 0, 0) & \text{otherwise.} \end{cases}
\end{aligned}$$

For solution $(\boldsymbol{p}', \boldsymbol{u}'_{\tilde{i}}, \boldsymbol{u}'_{\tilde{j}})$ of problem $\mathbf{P}(\tilde{i}, \tilde{j})$, we set

$$\boldsymbol{p}_i^{\tilde{i}\tilde{j}} := \boldsymbol{p}', \ \ \boldsymbol{u}_i^{\tilde{i}\tilde{j}} := \begin{cases} \bar{\boldsymbol{u}}_i & \text{if } i \in I \setminus \{\tilde{i},\tilde{j}\} \\ \boldsymbol{u}'_i & \text{if } i = \tilde{i} \text{ or } i = \tilde{j} \end{cases}$$

If $\phi(\boldsymbol{p}^{\tilde{i}\tilde{j}}, \boldsymbol{u}^{\tilde{i}\tilde{j}}) < \phi(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$, then $(\boldsymbol{p}^{\tilde{i}\tilde{j}}, \boldsymbol{u}^{\tilde{i}\tilde{j}})$ is a better solution.

In order to calculate $g_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t}, u_{\tilde{j}t})$, we must solve problem (4.2) three times for each time period. Therefore, we recommend calculating $g_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t}, u_{\tilde{j}t})$ approximately by the method described in Appendix A.2.

If we use the best admissible move strategy in two-unit local search, then the number of problems $\mathbf{P}(\tilde{i}, \tilde{j})$ in each iteration is $|I| \times (|I| - 1)/2$. Thus, the strategy is impractical. In the present paper, we consider only the first admissible move strategy. The algorithm of two-unit local search is as follows.

Two-unit Local Search (LS2)

Step 0: Let an initial feasible solution $(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$ be given.

Step 1: Set $I^0 := I^2$. Choose an unit pair $(\tilde{i}, \tilde{j}) \in I^0$.

Step 2: Solve problem $\mathbf{P}(\tilde{i}, \tilde{j})$, and obtain the candidate solution $(\boldsymbol{p}^{\tilde{i}\tilde{j}}, \boldsymbol{u}^{\tilde{i}\tilde{j}})$.

Step 3: If $\phi(\boldsymbol{p}^{\tilde{i}\tilde{j}}, \boldsymbol{u}^{\tilde{i}\tilde{j}}) < \phi(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}})$, then set $(\bar{\boldsymbol{p}}, \bar{\boldsymbol{u}}) := (\boldsymbol{p}^{\tilde{i}\tilde{j}}, \boldsymbol{u}^{\tilde{i}\tilde{j}})$, and go to Step 1.

Step 4: Set $I^0 := I^0 / \{(\tilde{i}, \tilde{j})\}$. If $I^0 = \emptyset$, then terminate. Otherwise, choose another unit pair $(\tilde{i}, \tilde{j}) \in I^0$ and go to Step 2.

5 Numerical Results

In this section, we show the results for the following three numerical experiments in order to examine the effect of the proposed methods.

1. Numerical experiments to examine the effect of the technique proposed in Subsection 3.2.2 for solving $\mathbf{ED}$(t) and (4.1).

2. Numerical experiments to compare with the existing methods [8, 12, 7] for the benchmark problems.

3. Numerical experiments to examine the behaviors of the proposed methods for the long-term UCP.

In numerical experiments 1 and 2, we solved the benchmark problems based on the 10-unit data and the 24-hour demand [10]. In numerical test 3, we solved the problems that are based on the 20-unit data and the four-week demand. The data are given in Table 5 and 6 and Figure 2-5. All numerical tests are implemented by Matlab 6.5 on a computer with a 2.53-GHz Pentium 4 CPU and 2.0 GB of memory.

5.1 Comparisons of the Quadratic Programming Solvers for $\mathbf{ED}(t)$ and (4.1)

We examine the effect of the technique proposed in Subsection 3.2.2 to solve quadratic programming problem $\mathbf{ED}(t)$. The proposed technique is applicable not only to problem $\mathbf{ED}(t)$ but also to problem (4.1) in the one-unit local search. We compare the computation times of the following two methods for $\mathbf{ED}(t)$ in LR and problem (4.1) in LS1.

- Technique proposed in Subsection 3.2.2: "proposed"
- General quadratic programming solver of Matlab: "quadprog"

We solved six UCPs, which are based on the data of Table 5. The 20, 40, 60, 80, and 100-unit data are created by duplicating the 10-unit base data. The load demands are multiplied by 2, 4, 6, 8, and 10 according to the number of units. The spinning reserve is assumed to be 10% of the demand in all cases.

The feasible solution was obtained by the LR method in Section 3 and LS1-best was executed. We set the parameters of (3.4) as $\epsilon = 0.002$ and $\sigma = 0.005$. We terminated the subgradient method if the number of iterations exceeds 100. We set the initial feasible solution of LS1-best as the LR-based solution and calculated the function.

Table 1 shows the total computation time of the LR method and LS1-best. The number in brackets indicates the computation time to solve problems **ED**(t) and (4.1).

Table 1: Effect of the technique proposed in Subsection 3.2.2 [s]

Unit	quadprog		proposed	
10	14.1	(11.5)	4.1	(1.5)
20	45.3	(40.3)	9.3	(4.3)
40	324.0	(312.4)	24.4	(12.9)
60	1232.8	(1212.2)	56.7	(36.1)
80	3699.1	(3666.2)	104.6	(71.7)
100	9220.6	(9170.0)	191.1	(140.4)

Table 1 shows that the computation time of the proposed technique is much less than that of quadprog. Since the number of variables of **ED**(t) or (4.1) is equal to that of on-units, the size of these problems increases with the number of units. Therefore, the computation time increases in proportion to the cube of the number of units if we solve these problems by the general quadratic programming solver. On the other hand, the increase in the computation time of the proposed technique is not significant. As a result, the proposed technique can solve even the large-scale (100-unit) problem, which takes more than two hours by quadprog, in practical time (only approximately three minutes).

5.2 Comparisons with the Existing Methods for the Benchmark Problems

In order to compare the proposed methods with the existing methods [8, 12, 7], we solved the problems presented in Subsection 5.1 using the existing methods. We set the parameters in (3.4) and the stopping criteria of the subgradient method for the dual problem as described in Subsection 5.1.

The proposed LS1-best, LS1-first, and LS2 were executed after the feasible solution was obtained by the LR method. We set the initial feasible solution of LS1-best and LS1-first as the LR-based solution, and set the initial feasible solution of LS2 as the solution obtained by LS1-best. We calculated $g_{\tilde{i}}(t; u_{\tilde{i}t})$ and $g_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t}, u_{\tilde{j}t})$ approximately using the methods described in the Appendix.

We list the total production costs in Table 2 and the computation time in Table 3. In Tables 2 and 3, "LRGA", "ELR", and "MILP" denote the LR method, in which genetic algorithms were incorporated [8], the LR method with various heuristics [12], and mixed-integer programming using linearized functions [7], respectively. The results of LRGA, ELR, and MILP are from the cited literature. Note that the computation times of LS1-first and LS1-best include the computation time of the LR method and that of LS2 includes the computation time of LS1-best.

Table 2: Comparison of Total Production Costs [$]

Units	LRGA[8]	ELR[12]	MILP[7]	LR	LS1-first	LS1-best	LS2
10	564800	563977	-	568356	564970	564970	563978
20	1122622	1123297	-	1129666	1125141	1125064	1123342
40	2242178	2244237	-	2256384	2244541	2242968	2242847
60	3371079	3363491	-	3378966	3365025	3361244	3360737
80	4501844	4485633	-	4501589	4483169	4482403	4481652
100	5613127	5605678	5605189	5627932	5602342	5600457	5599725

Table 2 shows that the proposed local search method greatly improves the LR-based feasible solution. The shaded cells in Table 2 indicate that the total production cost of the obtained solution is lower than that of the other methods. For the large-scale problems (60, 80, and 100 units), the proposed methods can obtain better solutions than the existing methods. Moreover, for the small problems (10, 20, and 40 units), the difference in costs by the proposed methods compared to the lowest-cost methods is quite small.

Table 3: Comparison of Computational Times [s]

Units	LRGA[8]	ELR[12]	MILP[7]	LR	LS1-first	LS1-best	LS2
10	518	4	-	2.2	2.5	2.8	12.7
20	1147	16	-	4.1	4.7	5.4	51.8
40	2165	52	-	8.1	10.3	13.5	135.9
60	2414	113	-	12.3	15.8	25.8	701.0
80	3383	209	-	16.7	25.8	39.7	1276.0
100	4045	345	123	21.3	36.7	61.9	2330.9

Next, we discuss the computation times presented in Table 3. As a result of the different experimental environment in [8], [12], [7], and the present paper, direct comparison of the computation times is meaningless. However, we can confirm the tendency of the algorithms as the scale of problems is increasing. Table 3 shows that the computation time of LS1 does not matter if the number of units is large. On the other hand, that of LS2 increases as the number of units increases. This happens because we have to solve the problem $\mathbf{P}(\tilde{i}, \tilde{j})$ $|I_2|$ times at each iteration and $|I_2| = O(|I|^2)$. Since the duality gap decreases as the number of units increases [5], we can expect that a solution by one-unit local search is sufficient when the number of units is large. The total production costs of the 60-, 80-, and 100-unit UCPs obtained by LS1-best are lower than those obtained by the existing methods, although, from the viewpoint of cost, LS2 is the best method.

5.3 Behavior of the Proposed Methods for Long-term Problems

We solved the problems with four different planning horizons ($T = 168, 336, 504, 672$). The number of units is 20, and the unit data is listed in Table 6. The demands of four different terms are created by connecting the data of Figure s2-5. For example, 336-hour demand is created by connecting the data of Figure s2 and 3. The spinning reserve is assumed to be 10% of the demand in all cases. Note that the startup cost is assumed to be constant S, that is, $S^{\text{hot}} = S^{\text{cold}}$.

We set the parameters of (3.4) as $\epsilon = 0.002$ and $\sigma = 0.005$, and terminated the subgradient method if the number of iterations exceeds 200.

Table 4 shows the results. In Table 4, "rate" denotes the proportion of each total production cost when the cost by LS2 is set to be 100. This indicates the degree of improvement from the LR-based solution by the proposed local search methods.

Table 4: Results of the long-term UCP

T		LR	LS1-first	LS1-best	LS2
168	cost[10^3yen]	2118560	2009532	1980779	1960603
	rate[%]	108.1	102.5	101.0	100.0
	time[s]	3.4	8.7	15.5	255.3
336	cost[10^3yen]	4387342	3989454	3966719	3882782
	rate[%]	113.0	102.7	102.2	100.0
	time[s]	6.3	21.4	41.4	716.4
504	cost[10^3yen]	6284472	5870819	5800511	5706400
	rate[%]	110.1	102.9	101.6	100.0
	time[s]	10.0	41.4	61.0	1222.6
672	cost[10^3yen]	8319286	7676362	7648401	7503018
	rate[%]	110.9	102.3	101.9	100.0
	time[s]	13.5	39.0	115.8	1398.3

Table 4 shows that LS1-first and LS1-best improve the LR-based solution greatly in all cases. Moreover, this table shows that LS2 improves the solution obtained by LS1 by more than 1%. Table 4 also shows that the computation times of the proposed methods increase almost linearly with the number of time periods. Note that the computation times of LRGA [8] and MILP [7] increase remarkably with the number of time periods. Since ELR [12] is specialized for the system of Table 5, ELR may be difficult to apply to other systems directly.

6 Conclusion and Future Research

In the present paper, we proposed new local search methods to improve the solution obtained by the Lagrangian relaxation method. We also proposed an efficient technique to solve the quadratic programming problems that arise in the proposed methods using its particular structure. The numerical results showed that the proposed methods are effective for large-scale UCPs, as compared with the existing methods. Moreover, the proposed methods are promising for long-term UCPs.

In the future, research will be conducted in order to examine more practical situations. For example, the ramp-rate limit constraint, which was removed for simplicity in the present paper, should be examined. Moreover, the proposed method might be extended to solve the system including not only thermal units but also pumped-storage units or area transitions.

Acknowledgment

First of all, I would like to express sincere appreciation to Associate Professor Nobuo Yamashita for his kind guidance, precise advice, and great help in program coding. I am thankful to Mr. Kaoru Kawamoto of Osaka Gas Co., Ltd for his cooperation in this research. I would like to tender my acknowledgement to Professor Masao Fukushima. His invaluable suggestions for my research and gentle behavior in daily life always encourage me. I also would like to express my thanks to Assistant Professor Shunsuke Hayashi for his appropriate comments and hard work. I would like to thank all members of Fukushima Laboratory, my teammates, and my friends. Finally, I really appreciate to Toto for his healing, and to my parents Mitsuo Seki and Yasuko Seki for their sincere support.

References

[1] E. H. L. Aarts and J. K. Lenstra, "Local search in combinatorial optimization", John Wiley and sons, 1997.

[2] J. F. Bard, "Short-term scheduling of thermal-electric generations using Lagrangian relaxation", Operations Research, Vol. 36, No.5, pp.756-766, 1988.

[3] M. S. Bazaraa and C. M. Shetty, "Nonlinear programming", John Wiley and Sons, 1979.

[4] D. P. Bertsekas, "Dynamic programming and optimal control", Athena Scientific, Belmont 1995.

[5] D. P. Bertsekas, D. S. Lauer, N. R. Sandell Jr, and T. A. Posbergh, "Optimal short-term scheduling of large-scale power Systems", IEEE Transactions on Automatic Cotrol, Vol. AC-28, No. 1, 1983.

[6] A. Borghrtti, A. Frangioni, F. Lacalandra and C. A. Nucci, "Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment", IEEE Transactions on Power System, Vol. 18, No.1, pp.313-323, 2003.

[7] M. Carrion and J. M. Arroyo, "A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem", IEEE Transactions on Power Systems, Vol. 21, No. 3, pp.1371-1378, 2006.

[8] C. P. Cheng, C. W. Liu, and C. C. Liu, "Unit commitment by Lagrangian relaxation and genetic algorithms", IEEE Transactions on Power Systems, Vol. 15, No. 2, pp.707-714, 2000.

[9] A. I. Cohen and M. Yoshimura, "A branch-and-bound algorithm for unit commitment", IEEE Transactions on Power Systems, Vol. 15, No. 2, pp.707-714, 2000.

[10] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, "A genetic algorithm solution to the unit commintment problem", IEEE Transactions on Power Systems, Vol. 11, No. 1, pp83-92, 1996.

[11] J. A. Muckstardt and S. A. Koenig, ”An application of Lagrangian relaxation to scheduling in power-generating systems”, Operations Research, Vol. 25, No. 3, pp387-403, 1977.

[12] W.Ongsakul, N. Petcharaks, ”Unit commitment by enhanced adaptive Lagrangian relaxation”, IEEE Transactions on Power Systems, Vol. 19, No. 1, pp.620-628, 2004.

[13] N. P. Padhy, ”Unit commitment - a bibliographical survey”, IEEE Transactions on Power Systems, Vol. 19, No. 2, pp1196-1205, 2004.

[14] G. B. Sheble, ”Solution to the unit commitment problem by the method oh unit periods”, IEEE Transactions on Power Systems, Vol. 5, No. 1, pp257-260, 1990.

[15] W. L. Snyder Jr, H. D. Powell Jr, and J. C. Rayburn, ”Dynamic programming approach to unit commitment”, IEEE Transactions on Power Systems, Vol. 2, pp339-350, 1987.

[16] S. Takriti and J. R. Birge, ”Using integer programming to refine Lagrangian-based unit commitment solutions”, IEEE Transactions on Power Systems, Vol. 15, No. 1 pp151-156, 2000.

[17] F. Zhuang and F. D. Galiana, ”Towards a more rigorous and practical unit commitment by Lagrangian relaxation”, IEEE Transactions on Power Systems, Vol. 3, No. 2, pp763-773, 1988.

A Approximate Methods for the Calculation of Costs in Local Search

A.1 An Approximate Method for the Calculation of $g_{\tilde{i}}(t; u_{\tilde{i}t})$

We propose an approximate method by which to calculate the function $g_{\tilde{i}}(t; u_{\tilde{i}t})$ defined by

$$g_{\tilde{i}}(t; u_{\tilde{i}t}) = \min \left\{ \sum_{i \in I \setminus \{\tilde{i}\}} \bar{u}_{it} f_i(p_{it}) + u_{\tilde{i}t} f_{\tilde{i}} p_{\tilde{i}t} \,\middle|\, \begin{array}{l} D_t = \sum_{i \in I \setminus \{\tilde{i}\}} \bar{u}_{it} p_{it} + u_{\tilde{i}t} p_{\tilde{i}t} \\ u_{\tilde{i}t} p_{\tilde{i}}^{\min} \leq p_{\tilde{i}t} \leq u_{\tilde{i}t} p_{\tilde{i}}^{\max} \\ \bar{u}_{it} p_i^{\min} \leq p_{it} \leq \bar{u}_{it} p_i^{\max}, \ \forall i \in I \setminus \{\tilde{i}\} \end{array} \right\}$$

Let I_t and I_t^{fix} be defined as

$$\begin{aligned} I_t &:= \{i \in I \setminus \{\tilde{i}\} \mid \bar{u}_{it} = 1\} \\ I_t^{\text{fix}} &:= \{i \in I \setminus \{\tilde{i}\} \mid \bar{u}_{it} = 0\}, \end{aligned} \tag{A.1}$$

respectively. Then, $g_{\tilde{i}}(t; u_{\tilde{i}t})$ can be rewritten as

$$g_{\tilde{i}}(t; u_{\tilde{i}t}) = \min \left\{ \sum_{i \in I_t^{\text{fix}}} \bar{u}_{it} f_i(\bar{p}_{it}) + \sum_{i \in I_t} f_i(p_{it}) + u_{\tilde{i}t} f_{\tilde{i}}(p_{\tilde{i}t}) \,\middle|\, \begin{array}{l} D_t = \sum_{i \in I_t^{\text{fix}}} \bar{u}_{it} \bar{p}_{it} + \sum_{i \in I_t} p_{it} + u_{\tilde{i}t} p_{\tilde{i}t} \\ u_{\tilde{i}t} p_{\tilde{i}}^{\min} \leq p_{\tilde{i}t} \leq u_{\tilde{i}t} p_{\tilde{i}}^{\max} \\ p_i^{\min} \leq p_{it} \leq p_i^{\max}, \ \forall i \in I_t. \end{array} \right\}$$

In order to obtain an approximate solution of the problem in the right-hand side, we consider relaxing the generation limit constraints $p_i^{\min} \leq p_{it} \leq p_i^{\max}, \ i \in I_t$. That is, we consider the following function:

$$g'_{\tilde{i}}(t; u_{\tilde{i}t}) = \min \left\{ \sum_{i \in I_t^{\text{fix}}} \bar{u}_{it} f_i(\bar{p}_{it}) + \sum_{i \in I_t} f_i(p_{it}) + u_{\tilde{i}t} f_{\tilde{i}}(p_{\tilde{i}t}) \,\middle|\, \begin{array}{l} D_t = \sum_{i \in I_t^{\text{fix}}} \bar{u}_{it} \bar{p}_{it} + \sum_{i \in I_t} p_{it} + u_{\tilde{i}t} p_{\tilde{i}t} \\ u_{\tilde{i}t} p_{\tilde{i}}^{\min} \leq p_{\tilde{i}t} \leq u_{\tilde{i}t} p_{\tilde{i}}^{\max} \end{array} \right\} \tag{A.2}$$

instead of $g_{\tilde{i}t}(t; u_{\tilde{i}t})$. To this end, we solve the equivalent minimization problem:

$$\begin{aligned} \min \quad & \sum_{i \in I_t} f_i(p_{it}) + u_{\tilde{i}t} f_{\tilde{i}}(p_{\tilde{i}t}) \\ \text{s.t.} \quad & D'_t = \sum_{i \in I_t} p_{it} + u_{\tilde{i}t} p_{\tilde{i}t} \\ & u_{\tilde{i}t} p_{\tilde{i}}^{\min} \leq p_{\tilde{i}t} \leq u_{\tilde{i}t} p_{\tilde{i}}^{\max}, \end{aligned} \tag{A.3}$$

where

$$D'_t = D_t - \sum_{i \in I_t^{\text{fix}}} \bar{u}_{it} \bar{p}_{it}.$$

The problem (A.3) is deduced from the minimization problem of the right-hand side of (A.2) by removing the constant term of the objective function and the constraints. The number of decision variables of problem (A.3) is $|I_t| + 1$. However, the variables $p_{it}, i \in I_t$ are represented by $p_{\tilde{i}t}$. Now, we consider the following problem:

$$\begin{aligned} \min \quad & \sum_{i \in I_t} (a_i p_{it}^2 + b_i p_{it} + c_i) \\ \text{s.t.} \quad & d_t = \sum_{i \in I_t} p_{it}. \end{aligned} \tag{A.4}$$

where $d_t = D'_t - p_{\tilde{i}t}$. Note that (A.4) is the problem with $p_{\tilde{i}t}$ fixed. The Karush-Kuhn-Tucker (KKT) conditions of the problem (A.4) are written as

$$2a_i p_{it} + b_i - \lambda = 0,\ \forall i \in I_t$$
$$\sum_{i\in I_t} p_{it} = d_t.$$

From the first equalities in the KKT condition, we have

$$p_{it} = \frac{\lambda - b_i}{2a_i},\ \forall i \in I_t.$$

From the second equality, we have

$$d_t = \sum_{i\in I_t} p_{it} = \sum_{i\in I_t} \frac{\lambda - b_i}{2a_i} = \lambda \sum_{i\in I_t} \frac{1}{2a_i} - \sum_{i\in I_t} \frac{b_i}{2a_i}.$$

Thus,

$$p_{it} = \frac{\alpha d_t + \beta - b_i}{2a_i},\ \forall i \in I_t,$$

where

$$\alpha = \frac{1}{\sum_{i\in I_t} \frac{1}{2a_i}},\quad \beta = \alpha \sum_{i\in I_t} \frac{b_i}{2a_i}.$$

is the optimal solution of the problem (A.4). Then, we have

$$\begin{aligned}\sum_{i\in I_t} a_i p_{it}^2 &= \frac{\alpha^2 d_t^2}{2} \sum_{i\in I_t} \frac{1}{2a_i} + \alpha\beta d_t \sum_{i\in I_t} \frac{1}{2a_i} - \alpha d_t \sum_{i\in I_t} \frac{b_i}{2a_i} + \sum_{i\in I_t} \frac{(\beta - b_i)^2}{4a_i} \\ &= \frac{\alpha}{2} d_t^2 + \beta d_t - \beta d_t + \sum_{i\in I_t} \frac{(\beta - b_i)^2}{4a_i} = \frac{\alpha}{2} d_t^2 + \sum_{i\in I_t} \frac{(\beta - b_i)^2}{4a_i}\end{aligned}$$

and

$$\begin{aligned}\sum_{i\in I_t} b_i p_{it} &= \sum_{i\in I_t} b_i \frac{\alpha d_t + \beta - b_i}{2a_i} \\ &= \alpha d_t \sum_{i\in I_t} \frac{b_i}{2a_i} + \beta \sum_{i\in I_t} \frac{b_i}{2a_i} - \sum_{i\in I_t} \frac{b_i^2}{2a_i} \\ &= \beta d_t + \beta \sum_{i\in I_t} \frac{b_i}{2a_i} - \sum_{i\in I_t} \frac{b_i^2}{2a_i}.\end{aligned}$$

Therefore, the optimal value of problem (A.4) is given by

$$\sum_{i\in I_t} (a_i p_{it}^2 + b_i p_{it} + c_i) = \frac{\alpha}{2} d_t^2 + \beta d_t + \sum_{i\in I_t} \frac{(\beta - b_i)^2}{4a_i} + \beta \sum_{i\in I_t} \frac{b_i}{2a_i} - \sum_{i\in I_t} \frac{b_i^2}{2a_i} + \sum_{i\in I_t} c_i. \tag{A.5}$$

Since $d_t = D'_t - p_{\tilde{i}t}$ in (A.5), we have

$$\begin{aligned}&\sum_{i\in I_t} (a_i p_{it}^2 + b_i p_{it} + c_i) \\ &= \frac{\alpha}{2} (D'_t - p_{\tilde{i}t})^2 + \beta (D'_t - p_{\tilde{i}t}) + \sum_{i\in I_t} \frac{(\beta - b_i)^2}{4a_i} + \beta \sum_{i\in I_t} \frac{b_i}{2a_i} - \sum_{i\in I_t} \frac{b_i^2}{2a_i} + \sum_{i\in I_t} c_i \qquad \text{(A.6)} \\ &= \frac{\alpha}{2} p_{\tilde{i}t}^2 - (\alpha D'_t + \beta) p_{\tilde{i}t} + C,\end{aligned}$$

where

$$C = \frac{\alpha(D'_t)^2}{2} + \beta D'_t + \sum_{i\in I_t} \frac{(\beta - b_i)^2}{4a_i} + \beta \sum_{i\in I_t} \frac{b_i}{2a_i} - \sum_{i\in I_t} \frac{b_i^2}{2a_i} + \sum_{i\in I_t} c_i.$$

Using (A.6), problem (A.3) is equivalent to the following problem:

$$\begin{array}{ll} \min & u_{\tilde{i}t}\{(a_{\tilde{i}} + \frac{\alpha}{2})p_{\tilde{i}t}^2 + (b_{\tilde{i}} - \alpha D'_t - \beta)p_{\tilde{i}t} + c_{\tilde{i}}\} + C \\ \text{s.t.} & u_{\tilde{i}t}p_{\tilde{i}}^{\min} \le p_{\tilde{i}t} \le u_{\tilde{i}t}p_{\tilde{i}}^{\max}. \end{array} \tag{A.7}$$

If $u_{\tilde{i}t} = 1$, then problem (A.7) is the convex quadratic programming with one variable and its minimum solution $p_{\tilde{i}t}^*$ is given by

$$p_{\tilde{i}t}^* = \max\left\{p_{\tilde{i}}^{\min},\ \min\left\{p_{\tilde{i}}^{\max},\ \frac{-(b_{\tilde{i}} - \alpha D'_t - \beta)}{2(a_{\tilde{i}} + \frac{\alpha}{2})}\right\}\right\}$$

Needless to say, $\tilde{p}_{\tilde{i}t} = 0$ if $u_{\tilde{i}t} = 1$.

Therefore, we have

$$g'_{\tilde{i}}(t; u_{\tilde{i}t}) = \begin{cases} \sum_{i\in I_t^{\text{fix}}} \bar{u}_{it} f_i(\bar{p}_{it}) + \{(a_{\tilde{i}} + \frac{\alpha}{2}(p_{\tilde{i}t}^*)^2 + (b_{\tilde{i}} - \alpha D'_t - \beta)p_{\tilde{i}t}^* + c_{\tilde{i}}\} + C & \text{if } u_{\tilde{i}t} = 1) \\ \sum_{i\in I_t^{\text{fix}}} \bar{u}_{it} f_i(\bar{p}_{it}) + C & \text{if } u_{\tilde{i}t} = 0. \end{cases}$$

Note that the equation does not include an optimization problem.

Relaxing the generation limit constraints might violate the constraint in the optimal solution of problem (A.3). Thus, in implementation, we may remove from (A.3) the units that would violate the constraints. Concretely, the units, which are $\bar{p}_{it} = p_i^{\min}$ or $\bar{p}_{it} = p_i^{\max}$ in time period t, are removed from (A.3). In other words, we replace the set defined in (A.1) with

$$\begin{aligned} I_t &:= \{i \in I \setminus \{\tilde{i}\} \mid p_i^{\min} < \bar{p}_{it} < p_i^{\max},\ \bar{u}_{it} = 1\} \\ I_t^{\text{fix}} &:= \{i \in I \setminus \{\tilde{i}\} \mid \bar{p}_{it} = p_i^{\min} \text{ or } \bar{p}_{it} = p_i^{\max} \text{ or } \bar{u}_{it} = 0\} \end{aligned}$$

A.2 An Approximate Method for the Calculation of $g_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t}, u_{\tilde{j}t})$

Here, we propose an approximate method by which to calculate the function $g_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t}, u_{\tilde{j}t})$ defined by

$$g_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t}, u_{\tilde{j}t}) = \min\left\{ \sum_{i\in I\setminus\{\tilde{i},\tilde{j}\}} \bar{u}_{it} f_i(p_{it}) + \sum_{i\in\{\tilde{i},\tilde{j}\}} u_{it}p_{it} \;\middle|\; \begin{array}{l} D_t = \sum_{i\in I\setminus\{\tilde{i},\tilde{j}\}} \bar{u}_{it}p_{it} \\ u_{it}p_i^{\min} \le p_{it} \le u_{it}p_i^{\max},\ \forall i \in \{\tilde{i},\tilde{j}\} \\ \bar{u}_{it}p_i^{\min} \le p_{it} \le \bar{u}_{it}p_i^{\max},\ \forall i \in I \setminus \{\tilde{i},\tilde{j}\} \end{array} \right\}$$

Let J_t and J_t^{fix} be defined as

$$\begin{aligned} J_t &:= \{i \in I \setminus \{\tilde{i},\tilde{j}\} \mid \bar{u}_{it} = 1\} \\ J_t^{\text{fix}} &:= \{i \in I \setminus \{\tilde{i},\tilde{j}\} \mid \bar{u}_{it} = 0\}, \end{aligned} \tag{A.8}$$

respectively. Then, $g_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t}, u_{\tilde{j}t})$ can be rewritten as

$$g_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t}, u_{\tilde{j}t}) =$$

$$\min \left\{ \sum_{i \in J_t^{\text{fix}}} \bar{u}_{it} f_i(\bar{p}_{it}) + \sum_{i \in J_t} f_i(p_{it}) + \sum_{i \in \{\tilde{i},\tilde{j}\}} u_{it} f_i(p_{it}) \;\middle|\; \begin{array}{l} D_t = \sum_{i \in J_t^{\text{fix}}} \bar{u}_{it}\bar{p}_{it} + \sum_{i \in J_t} p_{it} + \sum_{i \in \{\tilde{i},\tilde{j}\}} u_{it} p_{it} \\ u_{it} p_i^{\min} \leq p_{it} \leq u_{it} p_i^{\max},\ \forall i \in \{\tilde{i}, \tilde{j}\} \\ p_i^{\min} \leq p_{it} \leq p_i^{\max},\ \forall i \in J_t. \end{array} \right\}$$

To obtain the approximate solution of the problem in the right-hand side, we consider relaxing the generation limit constraint $p_i^{\min} \leq p_{it} \leq p_i^{\max},\ i \in J_t$. That is, we consider the function:

$$g'_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t}, u_{\tilde{j}t}) =$$

$$\min \left\{ \sum_{i \in J_t^{\text{fix}}} \bar{u}_{it} f_i(\bar{p}_{it}) + \sum_{i \in J_t} f_i(p_{it}) + \sum_{i \in \{\tilde{i},\tilde{j}\}} u_{it} f_i(p_{it}) \;\middle|\; \begin{array}{l} D_t = \sum_{i \in J_t^{\text{fix}}} \bar{u}_{it}\bar{p}_{it} + \sum_{i \in J_t} p_{it} + \sum_{i \in \{\tilde{i},\tilde{j}\}} u_{it} p_{it} \\ u_{it} p_i^{\min} \leq p_{it} \leq u_{it} p_i^{\max},\ \forall i \in \{\tilde{i}, \tilde{j}\}. \end{array} \right\}$$

instead of $g_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t})$. The function $g'_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t}, u_{\tilde{j}t})$ can be calculated in a manner similar to the calculation of $g'_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t})$ by solving the following problem:

$$\begin{aligned} \min \quad & \sum_{i \in \{\tilde{i},\tilde{j}\}} u_{it}\{(a_i + \frac{\alpha}{2})p_{it}^2 + (b_i - \alpha D'_t - \beta)p_{it} + c_i\} + \alpha u_{\tilde{i}t} u_{\tilde{j}t} p_{\tilde{i}t} p_{\tilde{j}t} + C \\ \text{s.t.} \quad & u_{it} p_i^{\min} \leq p_{it} \leq u_{it} p_i^{\max},\ \forall i \in \{\tilde{i}, \tilde{j}\}, \end{aligned} \tag{A.9}$$

where

$$\alpha = \frac{1}{\sum_{i \in J_t} \frac{1}{2a_i}},\ \ \beta = \alpha \sum_{i \in J_t} \frac{b_i}{2a_i},\ C = \frac{\alpha (D'_t)^2}{2} + \beta D'_t + \sum_{i \in J_t} \frac{(\beta - b_i)^2}{4a_i} + \beta \sum_{i \in J_t} \frac{b_i}{2a_i} - \sum_{i \in J_t} \frac{b_i^2}{2a_i} + \sum_{i \in J_t} c_i.$$

Therefore, we have

$$g'_{\tilde{i}\tilde{j}}(t; u_{\tilde{i}t}, u_{\tilde{j}t}) =$$

$$\begin{cases} \sum_{i \in J_t^{\text{fix}}} \bar{u}_{it} f_i(\bar{p}_{it}) + \sum_{i \in \{\tilde{i},\tilde{j}\}} \{(a_i + \frac{\alpha}{2})(p_{it}^*)^2 + (b_i - \alpha D'_t - \beta)p_{it}^* + c_i\} + \alpha p_{\tilde{i}t}^* p_{\tilde{j}t}^* + C & \text{if } u_{\tilde{i}t} = 1, u_{\tilde{j}t} = 1 \\ \sum_{i \in J_t^{\text{fix}}} \bar{u}_{it} f_i(\bar{p}_{it}) + \{(a_{\tilde{i}} + \frac{\alpha}{2})(p_{\tilde{i}t}^*)^2 + (b_{\tilde{i}} - \alpha D'_t - \beta)p_{\tilde{i}t}^* + c_{\tilde{i}}\} + C & \text{if } u_{\tilde{i}t} = 1, u_{\tilde{j}t} = 0 \\ \sum_{i \in J_t^{\text{fix}}} \bar{u}_{it} f_i(\bar{p}_{it}) + \{(a_{\tilde{j}} + \frac{\alpha}{2})(p_{\tilde{j}t}^*)^2 + (b_{\tilde{j}} - \alpha D'_t - \beta)p_{\tilde{j}t}^* + c_{\tilde{j}}\} + C & \text{if } u_{\tilde{i}t} = 0, u_{\tilde{j}t} = 1 \\ \sum_{i \in J_t^{\text{fix}}} \bar{u}_{it} f_i(\bar{p}_{it}) + C & \text{if } u_{\tilde{i}t} = 0, u_{\tilde{j}t} = 0, \end{cases}$$

where $(p_{\tilde{i}t}^*, p_{\tilde{j}t}^*)$ is the minimum solution of the problem (A.9). Note that if $u_{\tilde{i}t} = 1$ and $u_{\tilde{j}t} = 0$ (or $u_{\tilde{i}t} = 0$ and $u_{\tilde{j}t} = 1$), then $(p_{\tilde{i}t}^*, p_{\tilde{j}t}^*)$ is given by

$$p_{\tilde{i}t}^* = \max\left\{p_{\tilde{i}}^{\min},\ \min\left\{p_{\tilde{i}}^{\max},\ \frac{-(b_{\tilde{i}} - \alpha D'_t - \beta)}{2(a_{\tilde{i}} + \frac{\alpha}{2})}\right\}\right\},\ p_{\tilde{j}t}^* = 0.$$

Note also that if $u_{\tilde{i}t} = 1,\ u_{\tilde{j}t} = 1$, then $(p_{\tilde{i}t}^*, p_{\tilde{j}t}^*)$ is the solution of the problem:

$$\begin{aligned} \min \quad & \sum_{i \in \{\tilde{i},\tilde{j}\}} \{(a_i + \frac{\alpha}{2})p_{it}^2 + (b_i - \alpha D'_t - \beta)p_{it} + c_i\} + \alpha p_{\tilde{i}t} p_{\tilde{j}t} + C \\ \text{s.t.} \quad & u_{it} p_i^{\min} \leq p_{it} \leq u_{it} p_i^{\max},\ \forall i \in \{\tilde{i}, \tilde{j}\}. \end{aligned} \tag{A.10}$$

Problem (A.10) is QP with only two variables, and we can solve this problem easily.

In implementation, we removed the units that would violate the generation limit constraint. In other words, we replace the set (A.8) with

$$J_t := \{i \in I \setminus \{\tilde{i}, \tilde{j}\} \mid p_i^{\min} < \bar{p}_{it} < p_i^{\max},\ \bar{u}_{it} = 1\}$$
$$J_t^{\text{fix}} := \{i \in I \setminus \{\tilde{i}, \tilde{j}\} \mid \bar{p}_{it} = p_i^{\min} \text{ or } \bar{p}_{it} = p_i^{\max} \text{ or } \bar{u}_{it} = 0\}$$

B Problem Data for the Numerical Experiments

Table 5: System Data 1

	$p^{\max}$ [MW]	$p^{\min}$ [MW]	a [\$/MW^{2}h]	b [\$/MWh]	c [\$/h]	t^{up} [h]	t^{down} [h]	S^{hot} [\$/h]	S^{cold} [\$/h]	t^{hot} [h]	$\bar{v}$ [h]
Unit1	455	150	0.00048	16.19	1000	8	8	4500	9000	5	8
Unit2	455	150	0.00031	17.26	970	8	8	5000	10000	5	8
Unit3	130	20	0.00200	16.60	700	5	5	550	1100	4	-5
Unit4	130	20	0.00211	16.50	680	5	5	560	1120	4	-5
Unit5	162	25	0.00398	19.70	450	6	6	900	1800	4	-6
Unit6	80	20	0.00712	22.26	370	3	3	170	340	2	-3
Unit7	85	25	0.00790	27.74	480	3	3	260	520	2	-3
Unit8	55	10	0.00413	25.92	660	1	1	30	60	0	-1
Unit9	55	10	0.00222	27.27	665	1	1	30	60	0	-1
Unit10	55	10	0.00173	27.79	670	1	1	30	60	0	-1

Hour	1	2	3	4	5	6	7	8	9	10	11	12
Demand[MW]	700	750	850	950	1000	1100	1150	1200	1300	1400	1450	1500
Hour	13	14	15	16	17	18	19	20	21	22	23	24
Demand[MW]	1400	1300	1200	1050	1000	1100	1200	1400	1300	1100	900	800

Table 6: 20-unit Data

	$p^{\max}$ [MW]	$p^{\min}$ [MW]	a [10^3yen/MW^{2}h]	b [10^3yen/MWh]	c [10^3yen/h]	t^{up} [h]	t^{down} [h]	S [10^3yen]	$\bar{v}$ [h]
Unit 1	50.0	12.5	0.0154	1.28	35	48	48	50	48
Unit 2	50.0	12.5	0.0154	1.28	35	48	48	50	48
Unit 3	50.0	12.5	0.0154	1.28	35	48	48	50	48
Unit 4	100.0	25.0	0.0112	1.33	85	48	48	250	48
Unit 5	100.0	25.0	0.0085	1.65	80	48	48	150	48
Unit 6	100.0	25.0	0.0090	1.68	85	14	10	250	-10
Unit 7	100.0	25.0	0.0090	1.68	85	14	10	250	-10
Unit 8	100.0	25.0	0.0090	1.68	85	14	10	250	-10
Unit 9	200.0	50.0	0.0043	1.65	170	14	10	300	-10
Unit 10	200.0	50.0	0.0043	1.65	170	14	10	300	-10
Unit 11	300.0	75.0	0.0102	2.85	915	14	10	1800	-10
Unit 12	300.0	75.0	0.0104	2.89	935	14	10	1800	-10
Unit 13	300.0	75.0	0.0104	2.89	935	14	10	1800	-10
Unit 14	400.0	100.0	0.0074	2.77	1180	14	10	2700	-10
Unit 15	400.0	100.0	0.0074	2.78	1185	14	10	2700	-10
Unit 16	400.0	100.0	0.0021	1.64	335	48	48	750	48
Unit 17	400.0	100.0	0.0077	2.87	1230	14	10	2700	14
Unit 18	400.0	100.0	0.0077	2.87	1230	14	10	2700	14
Unit 19	600.0	150.0	0.0014	1.64	500	48	48	1900	48
Unit 20	600.0	150.0	0.0014	1.64	500	48	48	2000	48

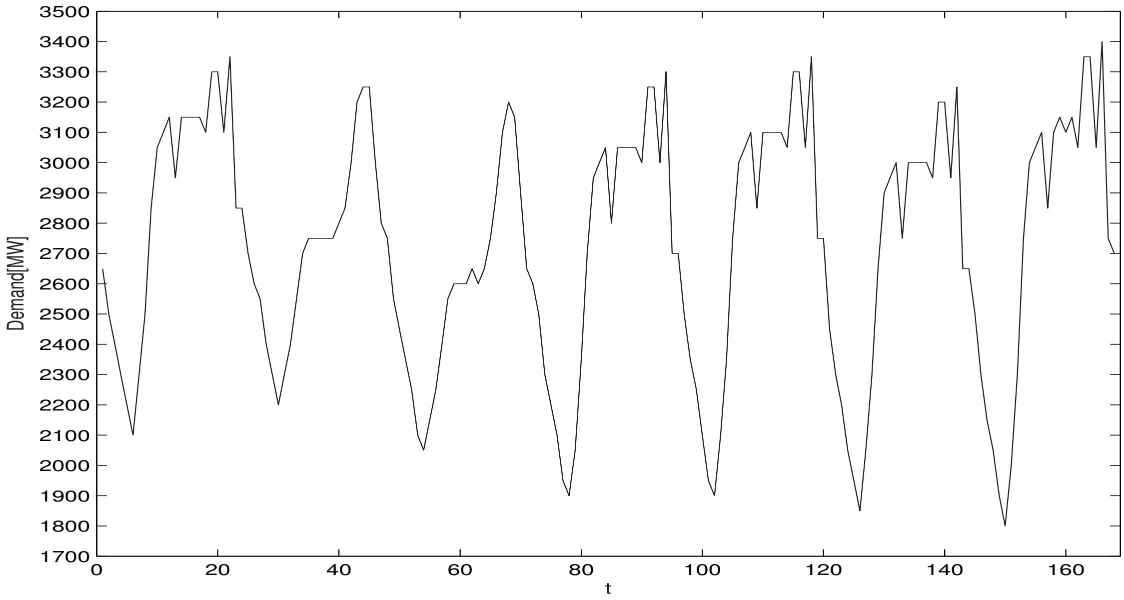

Figure 2: Demand in week 1 ($t = 1, \ldots, 168$)

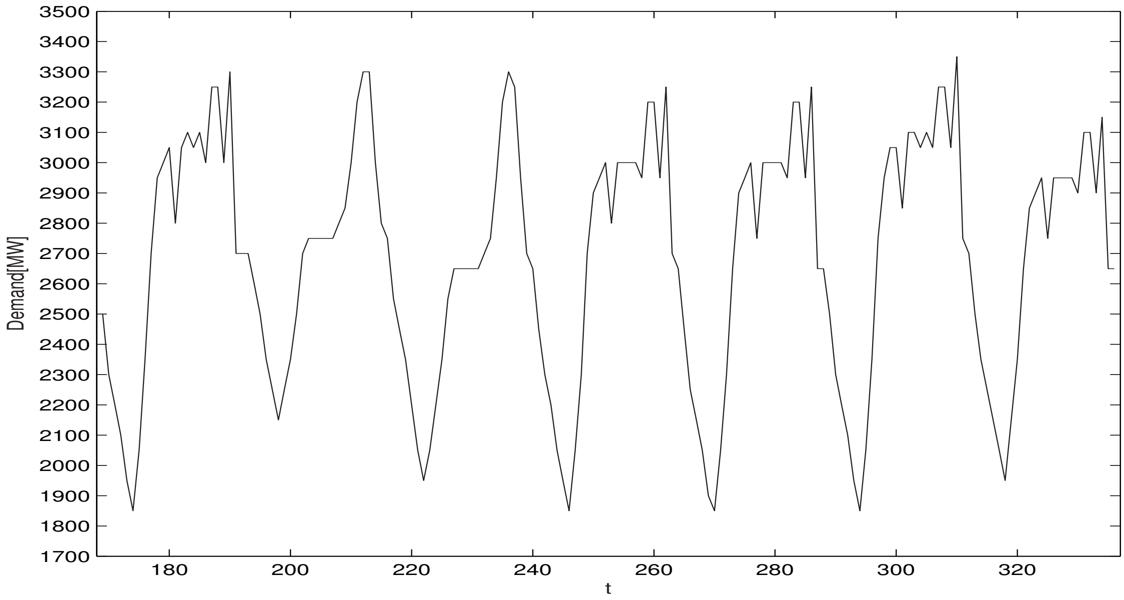

Figure 3: Demand in week 2 ($t = 169, \ldots, 336$)

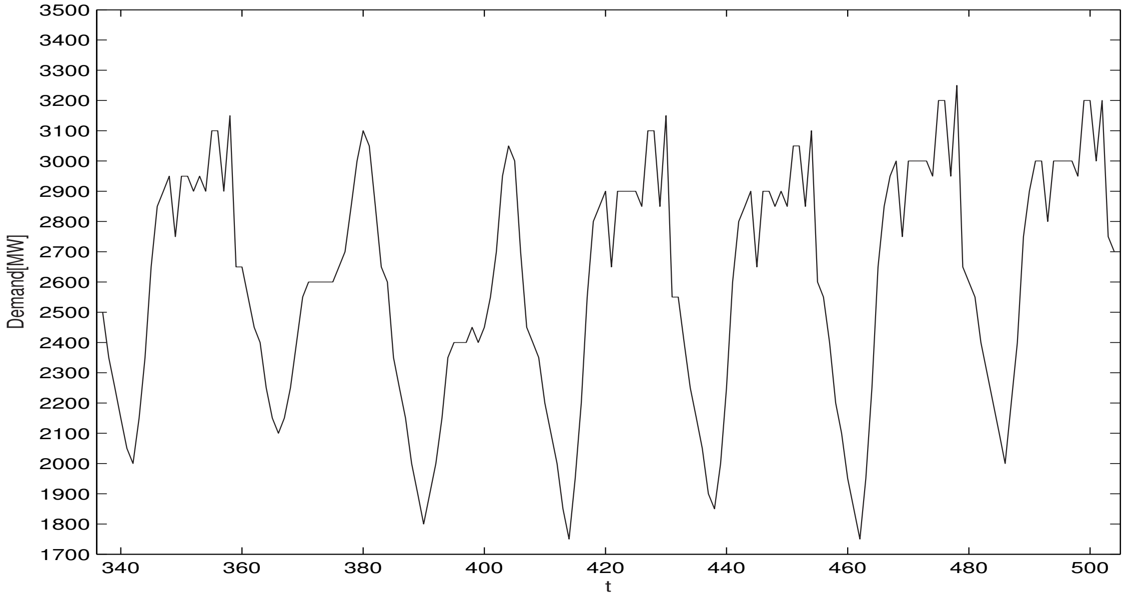

Figure 4: Demand in week 3 ($t = 337, \ldots, 504$)

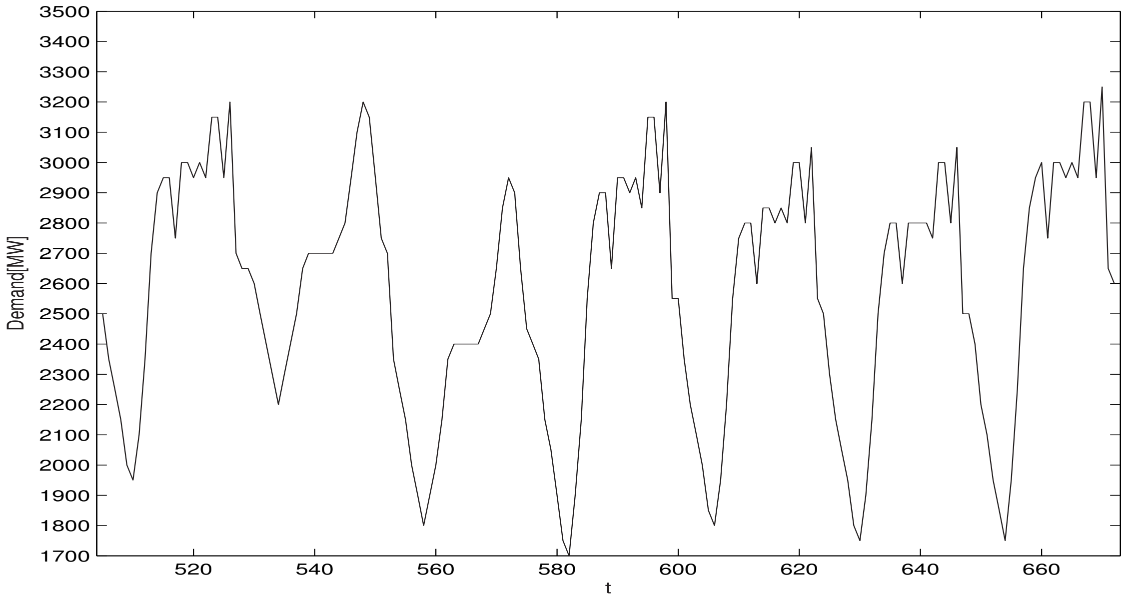

Figure 5: Demand in week 4 ($t = 505, \ldots, 672$)