An ADMM-based Approach for Traveling Salesman Problems

By Jun-hyon CHO[1)], Hyeok RYU[2)], and Jong-han KIM[1)]

[1)] *Department of Aerospace Engineering, Inha University, Incheon, Korea*

[2)] *Unmanned Aircraft System Research Division , Korea Aerospace Research Institute*

Traveling Salesman Problems (TSP) have been widely introduced for solving path planning or trajectory design problems. A typical TSP is described as follows; 1) a salesman is required to visit *n* cities with the shortest path and return to the place from which he left initially, and 2) each city must be visited only once. However, in fact, the TSPs are not limited to the problems of finding the shortest path of a whole travel, but it can be regarded as finding the safest travel that minimizes the integral of the risk function along the path. Hence the TSPs are extensively applied to mission planning or mission assignment problems in the field of robotics or aerospace systems.

There has been an extensive line of research works for solving the Integer Linear Programming problems into which the TSPs can be formulated. Popular examples include heuristic suboptimal approaches such as the greedy algorithms, the genetic algorithms, and the ant colony optimization algorithms, or more rigorous optimization-based approaches such as the convex relaxation via the semidefinite programming, or the branch-and-bound methods. In this paper, we reformulated the TSP as a Mixed Integer Linear Programming problem and applied the Alternating Direction Method of Multipliers (ADMM) on it. The ADMM, a kind of dual ascent methods, defines the augmented Lagrangian for strong convexity and achieve the fast convergence by coordinate-wisely minimizing the augmented Lagrangian. The main idea of the proposed method efficiently handling the integer constraints is as follows: the nonconvex integer constraints regarding the decision variable can be expressed as the set inclusion problem that the columns of the decision variable being included in the set of the unit vectors, and the proximal optimization step in the ADMM procedures can easily handle the set inclusion problem by computing the projection onto the finite set of the unit vectors. We also introduced a set of nonnegative slack variables to transform the inequality constraints of the Miller-Tucker-Zemlin (MTZ) formulation, for guaranteeing the connectivity of the travel, into equality constraints that can be efficiently handled. The convergence of the proposed method is investigated via computing the primal residuals and the dual residuals.

The performance of the proposed algorithm is verified via a series of numerical simulations on several well-known TSP examples. Sensitivity of the penalty parameters of the augmented Lagrangian term as well as the scalability of the proposed approach are investigated on a series of the numerical examples with different problem sized.

Key Words: Traveling Salesman Problem(TSP), Alternating Direction Method of Multipliers(ADMM), Nonconvex Optimization, Mixed Integer Linear Programming(MILP)

Nomenclature

N	:	the number of cities
$\mathcal{N}$	:	set of all cities $\{0, 1, \ldots, N-1\}$
$\mathcal{E}$	:	set of unit vectors
$\mathcal{S}$	:	set of matrices whose elements are between 0 and 1
$\mathbb{R}$	:	real space
$\mathbb{Z}$	:	integer space
$\circ$	:	Hadamard product
$\otimes$	:	Kronecker product
$\{0\}$	:	depot

1. Introduction

The Traveling Salesman Problem (TSP) is a combinatorial optimization problem of finding the shortest path for a salesman visiting all cities defined in a problem only once. Actually, the shortest path objective can be replaced by other meaningful criteria. Especially, for example in the field of aerospace engineering, when a UAV finds a route to spy on the enemy area, the cities can be replaced by the waypoints and the objective function can be the integral risk function modeled from the risk contour of enemy SAM sites.

The TSP is known as NP-hard, and there's no specific solution that solves it in polynomial time. But there have been a lot of studies that solve the optimal route of the problem. Deterministic algorithms, such as branch-and-bound, dynamic programming and cutting plane techniques, give exact solutions and they can deal with all TSP instances. But these are highly inefficient for some instances, so non-deterministic methods can be more useful giving approximate solutions.[1)] Because open-source solvers dealing with combinatorial optimization problems have used state-space search methods which inefficiency gives unfitness for implementation, an implemented algorithm is presented in this paper.

ADMM[2)] is an optimization method which solves the problems with a composite objective function. The ADMM is a dual ascent method, but it can be apart from it by introducing an augmented term which formulates the augmented Lagrangian and presents strong convexity that ensures faster convergence rate. Also, its proximal approach to the additional objectives is appropriate for implementation on parallel computing. In this paper, we relaxed the ILP to a mixed integer linear programming (MILP), and used the ADMM to solve it. A similar example was presented by Ref. 3), however the method is unsuitable for expanding the approach to the multi-agent TSP (mTSP) in the future because the solution of the mTSP is not in the set of Hamiltonian cycle. Our approach can be efficiently expanded to handle the mTSP by adopting LP relaxation to each salesman.

2. Taveling Salesman Problem as MILP

The TSP for N cities can be formulated as a integer linear programming (ILP) as follows.

$$\begin{aligned}
\text{minimize} \quad & \text{trace}\left(C^T X\right) \\
\text{subject to} \quad & \mathbf{1}^T X = \mathbf{1}^T, \ \ \mathbf{1}^T X^T = \mathbf{1}^T \\
& u_i - u_j + 1 \leq N\left(1 - X_{i,j}\right) \qquad i, j \in \mathcal{N}\backslash\{0\}, i \neq j \\
& \mathbf{0} \leq X_{i,j} \leq \mathbf{1}, \qquad i, j \in \mathcal{N} \\
\text{where} \quad & X \in \mathbb{Z}^{N\times N}, \quad u \in \mathbb{Z}^N
\end{aligned} \tag{1}$$

In the above problem, X is the decision variable where the (i, j) entry implying the existence of the path from city i to city j. The first constraint of the above problem means that the salesman should visit all cities once. The second line is the formulation of subtour elimination constraints called MTZ formulation.[4] When $X_{i,j}$ is equal to 1, the constraint for the path from city i to j becomes active, which means discontinuity of the route is not allowed. The other one is a valid constraint added for faster convergence.

To reformulate the TSP as a MILP, all variables are set as the real matrices or vectors, and rows of X must be in the set of unit vectors $\mathcal{E}$. Also, we introduced a slack variable Γ to the MTZ constraints to convert the inequalites to equalities for the ADMM. Then,

$$\begin{aligned}
\text{minimize} \quad & \text{trace}\left(C^T X\right) \\
\text{subject to} \quad & \mathbf{1}^T X = \mathbf{1}^T \\
& (Z_1)_i \in \mathcal{E}, \qquad i \in \mathcal{N} \\
& Z_2 \in \mathcal{S} \\
& N(Z_1)_{i,j} + u_i - u_j + \Gamma_{i,j} + 1 - N = 0 \\
& \qquad\qquad i, j \in \mathcal{N}\backslash\{0\}, i \neq j \\
& X - Z_1 = 0, \quad X - Z_2 = 0, \quad \Gamma \geq 0, \\
\text{where} \quad & X, Z_1, Z_2, \Gamma \in \mathbb{R}^{N\times N}, \quad u \in \mathbb{R}^N.
\end{aligned} \tag{2}$$

In the above equation, Z_1 and Z_2 are the variables that should be inside the set formed by inequalities in Prob. (1). Generally, the decision variable in equality constraints is not usually replaced by a new variable, but it is converted to Z_1 in this research.

3. NC-ADMM

Prob. (2) can be simplified as a set inclusion problem as follows.

$$\begin{aligned}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & Ax + Bz + c = 0 \\
& z \in C,
\end{aligned} \tag{3}$$

where the last line is the set inclusion constraint, and the set C can be either convex or nonconvex.

$$\begin{aligned}
\text{minimize} \quad & f(x) + \mathcal{I}_C(z) \\
\text{subject to} \quad & Ax + Bz + c = 0
\end{aligned} \tag{4}$$

In Prob. (4), $\mathcal{I}_C$ is the indicator function defined as

$$\mathcal{I}_C(x) = \begin{cases} 0 & \text{if } x \in C \\ \infty & \text{otherwise} \end{cases} \tag{5}$$

Then, the augmented Lagrangian of Prob. (2) is defined as follows.

$$\begin{aligned}
&\mathcal{L}_\rho(X, Z_1, Z_2, u, \Gamma, W_1, W_2, R, y) = \text{trace}(C^T X) \\
&\quad + \frac{\rho_1}{2}\left(\|X - Z_1\|_F^2 + \|X - Z_2\|_F^2\right) + \frac{\rho_2}{2}\|\mathbf{1}^T X - \mathbf{1}^T\|_F^2 \\
&\quad + \frac{\rho_3}{2}\sum_{i\in\mathcal{N}\backslash\{0\}}\sum_{j\in\mathcal{N}\backslash\{0\}}\left\{N(Z_1)_{i,j} + u_i - u_j + \Gamma_{i,j} + 1 - N\right\}_{i\notin j}^2 \\
&\quad + W_1(X - Z_1) + W_2(X - Z_2) + (\mathbf{1}^T X - \mathbf{1}^T)y \\
&\quad + \sum_{i\in\mathcal{N}\backslash\{0\}}\sum_{j\in\mathcal{N}\backslash\{0\}} R_{i,j}\left\{N(Z_1)_{i,j} + u_i - u_j + \Gamma_{i,j} + 1 - N\right\}_{i,j} \\
&\quad + \sum_{i\in\mathcal{N}} \mathcal{I}_\mathcal{E}(Z_1)_i + \mathcal{I}_\mathcal{S}(Z_2) + \mathcal{I}_{\mathbb{R}^+}(\Gamma)
\end{aligned} \tag{6}$$

In the general formulation of the augmented Lagrangian, single penalty parameter ρ is used for the augmented term. However, with the same penalty for all constraints, some constraints may not be satisfied because the level of residual for each constraint may not be quite different. For example, assuming a TSP with 10 cities, u should activate with the range of $(-5, 5)$ or $(0, 9)$, but X still operates with the range of $(0, 1)$. This would makes the first and the last constraint of Prob. (2) unsatisfied if the same penalty is used for them. So, multiple penalties were considered to formulate the Lagrangian, and the pareto optimal solutions were found.

The ADMM iteration can be derived from calculating a solution that the gradient of the augmented Lagrangian for each variable is equal to zero. The following statement tells the procedure of it.

X-update:

The gradient of X for Eq. (6) is

$$\begin{aligned}
\nabla_X \mathcal{L}_\rho =& C + \rho_1\left(2X - \sum_{i=1}^{2} Z_i\right) \\
& + \rho_2(\mathbf{1}\mathbf{1}^T X - \mathbf{1}\mathbf{1}^T) + \sum_{i=1}^{2} W_i + \mathbf{1}\otimes y^T,
\end{aligned} \tag{7}$$

and the solution that Eq. (7) is equal to zeros is

$$\begin{aligned}
X =& -(2\rho_1 I + \rho_2\mathbf{1}\mathbf{1}^T)^{-1} \\
& \left\{C - \rho_1\sum_{i=1}^{2} Z_i - \rho_2\mathbf{1}\mathbf{1}^T + \sum_{i=1}^{2} W_i + \mathbf{1}\otimes y^T\right\}.
\end{aligned} \tag{8}$$

Z-update:

Deriving the Z-update step should be separated for Z_1 and Z_2 because Z_1 is related to MTZ constraints. Then, first of all, the gradient of Z_1 is expressed as follows.

$$\begin{aligned}\nabla_{Z_1}\mathcal{L}_\rho =&\rho_1(-X+Z_1)+\partial\mathcal{I}_{\mathcal{E}}(Z_1)\\&+\rho_3 NF\circ\left(NZ_1+U+\Gamma+1-N+\frac{1}{\rho_3}R\right),\end{aligned} \tag{9}$$

where

$$\begin{aligned}U&=\mathbf{1}\otimes u^T-\mathbf{1}^T\otimes u\\F_{i,j}&=\begin{cases}0 & i,j\in\mathcal{N}\backslash\{0\}, i\neq j\\1 & \text{otherwise}\end{cases}.\end{aligned} \tag{10}$$

From Prob. (2), the MTZ formulation doesn't count of self-loops and a depot, but we counted them as assuming these are inactive and introduced F to be not updated over the iterations. This makes the implementation process easier with trading off memory capacity because all variables can be defined as the same size of matrices with the decision variable except for u and y. Back to the point, the solution is

$$\begin{aligned}z_1=&-\left[\rho_1 I+\rho_3 N^2\text{diag}\{\text{vec}(F)\}\right]^{-1}\\&\text{vec}\left\{-\rho_1 X-W_1+\rho_3 NF\circ\left(U+\Gamma+1-N+\frac{1}{\rho_3}R\right)\right\},\end{aligned} \tag{11}$$

where $z_1=\text{vec}(Z_1)$. Note that the subgradient of the indicator is not considered in the above, but it is implemented as a subgradient projection step after turning back z_1 to Z_1 as follow as

$$(Z_1)_i\leftarrow\mathbf{P}_{\mathcal{E}}\{(Z_1)_i\}. \tag{12}$$

$\mathbf{P}_{\mathcal{E}}$ is the projector on the nonconvex set $\mathcal{E}$, and the projection procedure is

$$\mathbf{P}_{\mathcal{E}}(v)=e_i,\qquad i=\text{argmax}\{v_1,\dots,v_N\}. \tag{13}$$

where e_i is the i-th unit vector in $\mathbb{R}^N$. Then, the update step for Z_2 is very simple as

$$\begin{aligned}&Z_2\leftarrow\mathbf{P}_{\mathcal{S}}\left(X+\frac{1}{\rho_1}W_2\right),\\&\text{where}\quad\mathbf{P}_{\mathcal{S}}(V)=\left\{W\;\middle|\;\begin{array}{ll}W_{i,j}=0, & V_{i,j}\leq 0\\W_{i,j}=V_{i,j}, & \text{otherwise}\\W_{i,j}=1, & V_{i,j}\geq 1\end{array}\right\}.\end{aligned} \tag{14}$$

Note that V and W are the $N\times N$ matrices.

u-update:

The gradient of u is simply divided to the parallel steps for updating each entry of u. Then,

$$\begin{aligned}\nabla_{u_i}\mathcal{L}_\rho =&\rho_3\left\{Nh(Z_1,i)+2(N-1)u_i-2\sum_{j\in\backslash\{0,i\}}u_j+h(\Gamma,i)\right\}\\&+h(R,i)\\\text{where}\quad& h(X,i)=\sum_{j\in\backslash\{0,i\}}\left\{(F\circ X)_{i,j}-(F\circ X^T)_{i,j}\right\},\end{aligned} \tag{15}$$

and the solution of $\nabla_{u_i}\mathcal{L}_\rho=0$ is the update step of u_i as shown in the following expression,

$$\begin{aligned}u_i=&-\frac{1}{2(N-1)}\cdot\\&\left\{Nh(Z_1,i)-2\sum_{j\in\backslash\{0,i\}}u_j+h(\Gamma,i)-\frac{1}{\rho_3}h(R,i)\right\}.\end{aligned} \tag{16}$$

Γ-update:

The last step for Γ is derived as follows.

$$\begin{aligned}\nabla_\Gamma\mathcal{L}_\rho =&\rho_3 F\circ\left(NZ_1+U+\Gamma+1-N+\frac{1}{\rho_3}R\right)+\partial\mathcal{I}_{\mathbb{R}_+}(\Gamma)\\\Gamma =&\mathbf{P}_{\mathbb{R}_+}\left\{-F\circ\left(NZ_1+U+1-N+\frac{1}{\rho_3}R\right)\right\},\\\text{where}\quad&\mathbf{P}_{\mathbb{R}_+}(V)=\left\{W\;\middle|\;\begin{array}{ll}W_{i,j}=0, & V_{i,j}\leq 0\\W_{i,j}=V_{i,j}, & \text{otherwise}\end{array}\right\}.\end{aligned} \tag{17}$$

Also, note that V and W are the $N\times N$ matrices.

Dual-ascent:

Updating the dual variables is the same as that of the dual ascent method, so

$$\begin{aligned}W_i&\leftarrow W_i+\rho_1(X-Z_i),\qquad\text{for } i=1,2\\R&\leftarrow F\circ\{R+\rho_3(NZ_1+U+\Gamma+1-N)\}\\y&\leftarrow y+\rho_2(X^T\mathbf{1}-\mathbf{1}).\end{aligned} \tag{18}$$

Primal-dual residuals:

We can find the solution from the ADMM iteration as stated above, but it's hard to check if it converges to a saddle point by tracking the objective trace(C^TX). Actually, since the augmented Lagrangian embraces the penalties of constraints, primal-dual residuals were observed over the iterations.

$$\begin{aligned}r_1^{k+1}&=X^{k+1}-Z_1^{k+1}\\r_2^{k+1}&=X^{k+1}-Z_2^{k+1}\\r_3^{k+1}&=F\circ(NZ^{k+l}+U^{k+1}+\Gamma^{k+1}+1-N)\\r_4^{k+1}&=\mathbf{1}^TX^{k+1}-\mathbf{1}^T\\r^{k+1}&=\begin{bmatrix}\text{vec}(r_1^{k+1})^T & \text{vec}(r_2^{k+1})^T & \text{vec}(r_3^{k+1})^T & r_4^{k+1}\end{bmatrix}^T\\s^{k+1}&=-\rho_1\sum_{i=1}^{2}(Z_i^{k+1}-Z_i^{k+1})\end{aligned} \tag{19}$$

From the above equations, the superscript k implies the current iteration number. And r_i and s are the primal residual of each constraint and the dual residual respectively. Also, dual feasibility for X^{k+1} is represented as

$$C+\sum_{i=1}^{2}W_i^{k+1}+\mathbf{1}\otimes y^T. \tag{20}$$

As stated above, we dropped the dual residuals for the other variables except Z_1, since subgradients of the indicators cancel out the other terms of gradients of the dual ascent form. So, they are ignored in this paper.

By summing up all processes stated above, the proposed algorithm is organized as a pseudo code in a tabular form as follows. Note that all variables are updated simultaneously as each step is done like the Gauss-Seidel method. Also, the iteration terminates when the primal-dual residuals are small enough.

Algorithm 1 NC-ADMM based method for TSP

Require: C

Declaration $X^0, Z_1^0, Z_2^0, u^0, \Gamma^0, W_1^0, W_2^0, R^0, y^0, r^0, s^0$
Initialization $k = 0$
while $\|r^k\| \geq 1e-3$ **or** $\|s^k\|_F \geq 1e-1$ **do**
  $X^{k+1} \quad \leftarrow \quad (8)$
  $Z_1^{k+1} \quad \leftarrow \quad (11), (12)$
  $Z_2^{k+1} \quad \leftarrow \quad (14)$
  $u_i^{k+1} \quad \leftarrow \quad (16)$ **for** $i = 1, 2, \cdots, N$
  $\Gamma^{k+1} \quad \leftarrow \quad (17)$
  $W_1^{k+1}, W_2^{k+1}, R^{k+1}, y^{k+1} \quad \leftarrow \quad (18)$
  $r^{k+1}, s^{k+1} \quad \leftarrow \quad (19)$
  $k \quad \leftarrow \quad k+1$
end while

4. Numerical Experiments

4.1. Symmetric Examples

We first solved a famous 5-city problem[4)] where the cost matrix is represented in Table 1. The diagonal entries are not defined here, but we filled the entries with summation of the other entries, since the self-loops should not be selected. The cost matrix is divided by the maximum number of all entries before the beginning of the iteration, as a measure of normalization, since the algorithm is sensitive to the penalties.

Table 1. Cost matrix of the 5-city problem.

-	3	4	2	7
3	-	4	6	3
4	4	-	5	8
2	6	5	-	6
7	3	8	6	-

Table 2 is a result of the proposed method with $\rho_1 = 1, \rho_2 = 1$, and $\rho_3 = 0.1$. The result is identified as the optimal solution by comparing to the same result of an open-source solver (CVXPY-GLPK) using the branch-and-bound method.

Table 2. Result of the 5-city problem.

0	0	1	0	0
0	0	0	0	1
0	1	0	0	0
1	0	0	0	0
0	0	0	1	0

Figure 1 shows the trend of primal-dual residuals over the iterations. From Figure 1, dual feasibility is valid showing that it's the same with dual residual. Figure 2 displays that the residual for violating the MTZ constraints is quite smaller than the other residuals, even ρ_3 is only 10% of ρ_1 and ρ_2. If the same penalties are used, the other constraints except the subtour elimination condition will not be satisfied through the iterations.

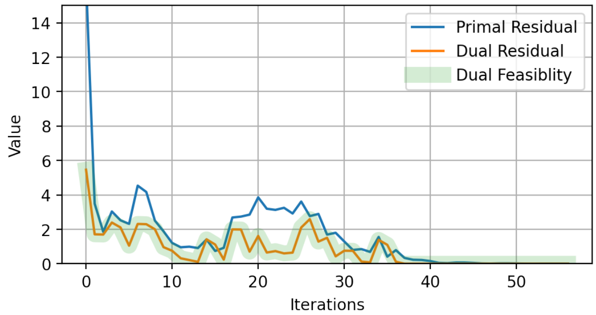

Fig. 1. Primal-dual residuals and dual feasibility through the iterations.

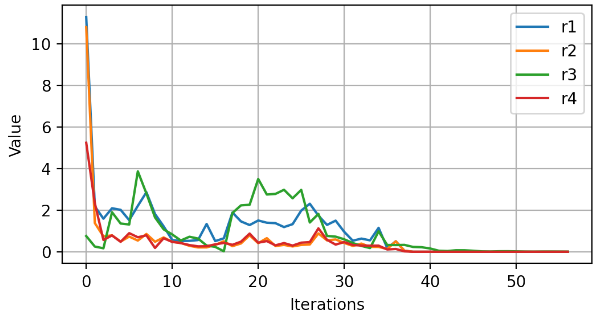

Fig. 2. Residuals for all constraints through the iterations.

The second experiment expands to the Barachet's 10-city problem.[5)] With $\rho_1 = 0.0885$, $\rho_2 = 0.0885$, and $\rho_3 = 0.00885$, the result and its convergence trend are displayed in Table 3 and Fig. 3, and these are verified with the same manner of the previous example.

Table 3. Result of the Barachet's 10-city problem.

						1			
1									
	1								
		1							
			1						
							1		
					1				
								1	
									1
				1					

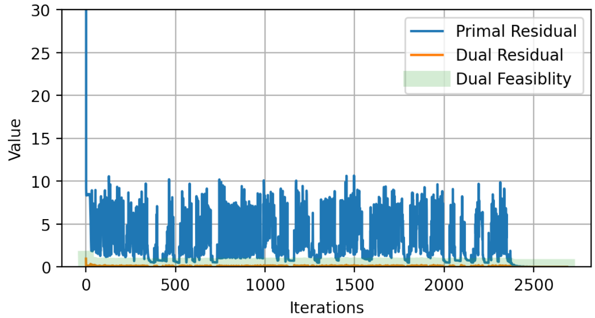

Fig. 3. Convergence trend for the 10-city problem.

4.2. Asymmetric Example

An experiment for the asymmetric case is also conducted. A random integer matrix is generated for the experiment and Table 4 is the cost matrix of the example. The optimal solution is represented in Table 5 and the algorithm converges through the iterations as shown in Fig. 4.

Table 4. Cost matrix of the asymmetric example.

-	16	38	99	9	75	75	27	10	7
76	-	83	40	97	96	31	95	11	25
48	36	-	90	46	69	11	24	7	84
35	40	31	-	71	81	72	7	15	56
5	98	98	99	-	16	35	31	76	56
33	29	96	51	67	-	76	41	5	63
48	69	24	63	50	88	-	66	14	34
57	48	92	37	24	52	74	-	11	70
41	89	41	49	58	83	9	91	-	24
42	35	67	49	48	40	32	55	63	-

Table 5. Result of the asymmetric example.

									1
			1						
	1								
							1		
1									
								1	
		1							
				1					
						1			
					1				

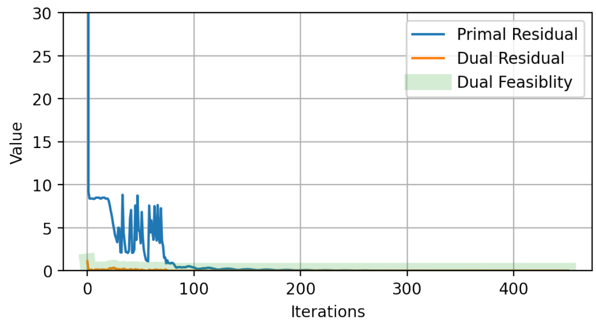

Fig. 4. Convergence trend for the asymmetric example.

5. Conclusion

In this paper, we presented an approach to solve the TSP via the ADMM. The nonconvex constraints appearing in the reformulation process can be handled by projecting onto the nonconvex set. The proposed method is validated by solving the symmetric and asymmetric examples. In the ADMM iteration for TSP, there are very few of matrix multiplications. Even the updating X and Z_1 is the most complicated step in the algorithm, the inverse matrices are very simple and any additional algorithms to solve linear systems are not required. Also, the algorithm is highly parallelized for easy implementation on embedded computers, which expands the range of applications on engineering fields.

As a future work, the TSP for multiple salesman which is more practical in engineering problems will be considered. For the multiple TSPs (mTSPs), the analysis of the penalties should be extensively demanded because constraints of the multiple agent case is way more complicated. Due to the difficulties for deciding the best penalties, finding the suboptimal solutions can be considered with trading off the performances.

Acknowledgments

This research was supported by Unmanned Vehicles Core Technology Research and Development Program through the National Research Foundation of Korea(NRF) and Unmanned Vehicle Advanced Research Center(UVARC) funded by the Ministry of Science and ICT, the Republic of Korea(No.: NRF-2021M3C1C1A02099428).

References

1) Goyal, S.: A Survey on Travelling Salesman Problem, *43rd proceeding of MICS*, USA, Jan. 2010.
2) Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J.: Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers, *Foundations and Trends in Machine Learning*, **3**(2011), pp.1–122.
3) Takapoui, R.: The Alternating Direction Method of Multipliers for Mixed-Integer Optimization Applications, Ph.D. Thesis, Stanford University, 2017.
4) Miller, C. E., Tucker, A. W. and Zemlin, R. A.: Integer Programming Formulation of Traveling Salesman Problems, *Journal of the ACM*, **7**(1960), pp.326–329.
5) Dantzig, G. B., Fulkerson, D. R. and Johnson, S. M.: On a Linear Programming, Combinatorial Approach to the Traveling-Salesman Problem, *Operations Research*, **7**(1959), pp.58-66.