Hybrid PSO-tabu search for constrained non-linear optimization problems

Abdelghani Bekrar[1], Sondes Chaabane[1], Damien Trentesaux [1], Augusto Bornschlegell[2], Julien Pellé[2], Souad Harmand[2]

[1] Université de Valenciennes, TEMPO/PSI
Le Mont Houy, 59313, Valenciennes cedex 9, France
abdelghani.bekrar@univ-valenciennes.fr, sondes.chaabane@univ-valenciennes.fr, Damien.Trentesaux@univ-valenciennes.fr

[2] Université de Valenciennes, TEMPO/DF2T
Le Mont Houy, 59313, Valenciennes cedex 9, France
augusto.s.b@gmail.com, jpelle@univ-valenciennes.fr, Souad.Harmand@univ-valenciennes.fr

Abstract

In this paper we present a new method to solve a constrained non-linear problem. The method is based on hybridizing the Particle Swarm Optimization and tabu-search meta-heuristics (PSO-TS). Tow tabu-lists are used within the PSO algorithm: the first one aims to diversify the best solutions obtained by particles when the second bans temporarily solutions non-respecting the constraints. The obtained meta-heuristic is validated on real thermal problem called T-junction problem. It consists on optimizing the thermal management of the system and minimizing its over heating by improving its design and the flow distribution. Our results are compared with Genetic algorithm.

Key words

Constrained non-linear optimization, Particle Swarm Optimization (PSO), Tabu-search (TS), hybrid method, flow and heat transfer optimization.

1 Introduction

Non-linear optimization problems are defined by non-linearity constraints and/or non-linearity objective. These problems are considered in several domains, including chemical engineering, energy analysis, environmental planning, biotechnology and thermal processes, among others. Different techniques and methods are employed to model and solve these problems.

A literature survey shows that the most used techniques are evolutionary algorithms [1, 2], swarm optimization [6] and non-linear mathematical programming [15]. Leyffer and Mahajan (2010) present a survey of non-linearly constrained software and methods, focusing on the contrasting strategies of local optimization and global optimization [15]. Some of those approaches such as Genetic algorithms are reported to require a lot of parameters and to entail considerable effort to implement.

In the thermal engineering field, many complex optimization problems arise in practice. Recently, non-linear optimization problems have increasingly been subjected to analysis by non-traditional optimization techniques. Patel and Rao [17, 18] recommend the use of particle swarm optimization (PSO) based on case studies showing that PSO is simple in concept, requires few parameters, is easy to implement and performs well compared to traditional techniques like genetic algorithms [17, 18].
The PSO method has produced good outcomes for a variety of optimization problems, but many authors have pointed out a limitation in its ability to diversify the population (see [8, 24]). To deal with this problem, research efforts are underway on several fronts to hybridize the PSO method with other

meta-heuristics.

The most commly used methods to create PSO hybrids are genetic algorithms and differential evolution algorithms [24]. For global optimization, a PSO-TS hybrid algorithm which joins PSO with tabu search (TS) has been proposed in [11]. More recently, Shelokar et al. hybridize PSO with ant colony algorithm for continuous optimization [21].

In this work, we focus on a thermal optimization problem known as the T-junction problem, which consists in designing the main channel in electrical machines responsible for evacuating generated heat. The objective is to determine the ideal channel features that optimize the temperature in the system. This problem, identified through a collaborative industrial project, can be formulated as a constrained non-linear optimization problem (CNOP). The fitness function used to evaluate solutions of this problem takes extensive computation time. The use of meta-heuristics like genetic algorithms in this case has proved to be very time consuming.

We apply the PSO meta-heuristic to solve the problem due to its simple implementation and the limited number of parameters to adjust, as well as for the ability to control its fitness function effectively. To avoid premature convergence of our method, a tabu search procedure is embedded within the PSO.

In the following section, we present the PSO meta-heuristic and the hybrid method called PSO-TS applied to the CNOP problems. In section 3 we explain the thermal problem and the general framework PSO-TS applied to it. Section 4 is dedicated to simulation and experimental results.

2 Hybrid PSO-Tabu search method

2.1 Particle Swarm optimization for non-linear optimization problem

The particle swarm optimization (PSO) method is a meta-heuristic proposed the first time by R.C. Eberhart and J. Kennedy in 1995 [9, 14]. Their main idea is to evolve initial solutions (particles) in order to find the best one. This evolution is done by analogy to the behavior of some species as they look for food, like a flock of birds or a school of fish ([9,14]). One can summarize the link between the species' behavior and the optimization process as follows [6]:

- Each solution is considered as a particle “flying” in the search space;
- To “fly”, particles have a velocity vector and direction of moving;
- Each particle can request information from other particles, called informants;
- A particle is able to move from its current position to a new one by modifying its velocity. The widely-used formula for updating a particle's position ([13]) is:

$$v_d(i, t+1) = w * v_d(i,t) + c_1 * rand() * (p_d(i) - x_d(i,t)) + c_2 * rand() * (g_d - x_d(i,t)) \quad (1)$$

$$x_d(i,t+1) = x_d(i,t) + v_d(i,t+1) \quad (2)$$

where:

- $x_d(i,t)$, the d^{the} component of the i^{the} current particle position at the iteration t,
- $v_d(i,t)$, its current velocity,
- p_d, the best position of the current particle achieved so far,
- g_d, the global best position achieved by all informants.
- w, an inertia weight introduced in [23], where c_1 and c_2 are randomly generated and called acceleration coefficients.

The new position of a particle depends on its velocity, which is modified by reference to its own best performance and by reference to the global best performance (over all informants).

The efficiency of PSO has been documented on a wide range of optimization problems. However, constrained non-linear problems have not been widely studied by this method. The chief effort to adapt PSO to the constrained non-linear area is that of Hu and Eberhart [4]. This modified PSO starts with initial particles that satisfy all the constraints (feasible solutions). On the evolution steps, the feasibility of particles is maintained by selecting only feasible particles as a source of the p_d and p_g parameters. The method is tested on constrained non-linear optimization instances and compared with a genetic algorithm.

By contrast, many other methods based on stochastic search have been proposed to tackle CNOP. The difficulty in adapting these different meta-heuristics mainly involves the question of how to preserve the feasibility of solutions during different iterations.

A variety of approaches can be used to deal with feasibility in CNOP which fall chiefly in two classes:

1. Penalty function approaches,
2. Approaches preserving feasibility throughout evolutionary computation,

The penalty function methods transform the constrained non-linear problem into a sequence of unconstrained optimization problems. The violated constraints in the unconstrained problem are penalized in the objective function with a penalty factor. The penalty functions can be stationary or non-stationary.

The second approach tries to preserve feasibility at each iteration. In those techniques, infeasible solutions are rejected, repaired or replaced by feasible solutions.

In our adaptation of the PSO to CNOP, we choose to use the penalty function approach because it is simple to implement. If the new position calculated by the inequality given above ((1) and (2)) is infeasible, a large penalty weight is assigned to the fitness function of this particle.

To accelerate the algorithm, the swarm is initialized with feasible solutions only. We generate particles randomly, and terminate the process when a population with a desired size respecting all the constraints is obtained.

The thermal problem constitutes a class of problems that is exceedingly hard to solve (see section 3.1). The objective function is non-linear and consumes substantial computing time to evaluate. To adapt PSO to this problem class, the call of the objective function is limited to feasible solutions. Details of our adapted PSO are sketched in Algorithm 1.

Algorithm1. PSO adapted for "T-junction" problem

- *n;* // swarm size= 20, 30, … 50
- *bound; //bound of variables*
- *X; //D-dimension vector of positions*
- *v; //velocity vector*

1. Initializing population *pop*(*n* solutions)
   a. **Repeat**
      i. Generate a solution *S* randomly, *S = rand(bound) ;*
      ii. If *S* is feasible, then add *S* to population,
      Pop(i) = S ; v(i) = rand(bound);
   b. **Until size of *pop* = *n***
2. **Repeat**
   1. For each particle i = 1… n
      1. **if** solution *i* is infeasible, then *eval(i) = max_value*
      2. **else** Evaluate the fitness of the particle: *eval(i),*
      3. **If** *eval(i)* is better than $p_d(i)$, then update $p_d(i)$,
   2. Update g_d

   // Updating velocity and position

   3. **For** each particle *i = 1… n*, **for** each dimension *d = 1… D*
      1. $v_d(i, t+1) = w * v_d(i,t) + c_1*rand()*(p_d(i) - x_d(i,t))+c_2* rand()*(g_d - x_d(i,t))$
      2. $x_d(i,t+1) = x_d(i,t) + v_d(i,t+1)$
3. **Until** *(stop criterion)*

In our implementation of PSO, parameters are adapted from [5] and [25]. More details are given in section 3.4.

2.2 Hybridization of PSO with Tabu Search (PSO-TS)

In order to avoid the premature convergence of the PSO method toward poor quality local optima, many papers proposed its hybridization with other methods. As observed in the review by Thangaraj et al. (2011), most of the proposed methods use genetic algorithms or differential evolution algorithms [24]. Only two papers using tabu search with PSO are cited. Moreover, most of the hybrid methods using tabu-search are proposed to solve discrete optimization problems, especially job shop scheduling problems (see [20, 22]).

The Tabu search algorithm is an efficient meta-heuristic proposed by Glover in 1986 [12] which makes use of adaptive memory processes for guiding the search. The simplest of these processes consists of recording features of visited regions of the search space in a *tabu list*, which provides a means to avoid revisiting solutions already encountered and so avoid becoming trapped in local optimality. This prohibition is temporary since it depends on the length of the tabu list (called tabu tenure).

In our framework, we use double tabu lists within the PSO:

- The first tabu list, named *infeasible_list*, is a supplementary list to facilitate the handling of infeasible solutions. We record infeasible solutions in a list with fixed size. When a new particle position is determined by equations (1) and (2), we check if it belongs to *infeasible_list.* In this case, the position is not updated, the precedent position is recovered.
  The aim in using the *infeasible_list* is to avoid checking the feasibility of solutions.
- The second tabu list is used in the diversification procedure detailed in the following section.

2.2.1 Diversification procedure

To deal with the risk of losing the diversity of solutions after many iterations of PSO, a tabu search procedure is embedded within PSO. This procedure serves as local optimizer of the best local solutions (*pbest*).

Tabu search was adapted for global optimization in many papers. However, our procedure is inspired by the adaptation of Chelouah and Siarry [4], called Enhanced Continuous Tabu Search (ECTS), which has obtained good results on known test functions.

The *pbest* solutions of PSO are an input to the procedure. For each solution *s*, a list of neighborhoods is defined. Candidate solutions from these neighborhoods are examined and the best one becomes the new current solution that replaces s. The solution s is saved in a second tabu list, called *best_list*. The process is repeated to produce successive new solutions until a defined stopping criterion is satisfied.

The neighborhoods of a solution *s* are defined by hyperrectangles introduced in [4]. A hyperrectangle of *s* with radius *r* is the space containing solutions *s*' such that the distance between *s* and *s' is* less than *r*.

To generate the *m* neighborhoods for the solution s, *m* hyperrectangles centered on s are created, and a point is randomly chosen from each. The best of the *m* chosen points then becomes *s*.

The diversification procedure is sketched in Algorithm 2. This pseudo-code represents an adaptation for the thermal problem studied in this paper.

The complete PSO-TS method is presented in a programming flow chart in the next section.

Algorithm2. Diversification procedure

Inputs

- *pbest;* // best historical solution of particles
- *pbestval;* solutions values
- *m; //neighbourhood size*
- *r; //radius of hyperrectangles*
- *eps; //threshold for accepting new solution*

1. *best_list = pbest;* // Initializing tabu list *best_list*
2. **Repeat**

    4. **For** each solution *s* in *pbest*

    //generation ***m*** neighbourhood

    While *i* < *m*

    1. *h1 = rand(pbest(i)[x1]-0.5*i*r, pbest(i)[x1]+0.5*i*r);*
    2. *h2 = rand(pbest(i)[x2]-0.5*i*r, pbest(i)[x2]+0.5*i*r);*
    3. **if** *h1 et h2* are in the previous hyperrectangles, **then** ensure that *h3* is out of;

        else *h3 = rand(pbest(i)[x3]-0.5*i*r, pbest(i)[x3]+0.5*i*r);*
    4. **if** *h=(h1,h2,h3)* $\notin$ *best_list* **then**
        1. add *h* to *best_list;*
        2. *i = i + 1;*
        3. **if** *eval(h)-pbestval(i)* $\leq$ *eps* **then** update *pbestval* and *pbest*

Until (*stop criteria*)

3 Solving a thermal problem by hybrid PSO-TS

In this section we present the application of our PSO-TS method to a thermal problem from the domain of constrained non-linear optimization problems.

3.1 The T-junction thermal problem

We consider a two-dimensional T-junction that has one air inlet (Q_{in}) at the main channel, as shown in the Figure (1.a). T-junctions are widespread in cooling systems of many industrial applications, for instance in electrical machines used in turbo alternators, railway transport, aircraft electric generators and so forth. In this study, the T-junction is defined by four variables, x_1 to x_4, for the given global dimensions (L, H). These variables correspond respectively to the T-junctions' center coordinates, the main channel width and the lateral branch width. The inlet fluid flow Q_{in} is divided into two portions at the T-junctions' center, the first keeps the flow moving straight ahead through the main channel and the second one turns the flow to the lateral branch. The flow distribution and the inlet mass flow rate depend on the head losses that are generated along the main channel and the lateral branch. The air flow is responsible for evacuating heat that is generated in the solid media (steel), which is thermally insulated from its surroundings. The thermal mathematical model is defined by the heat equation in Cartesian coordinates and steady state, as follows:

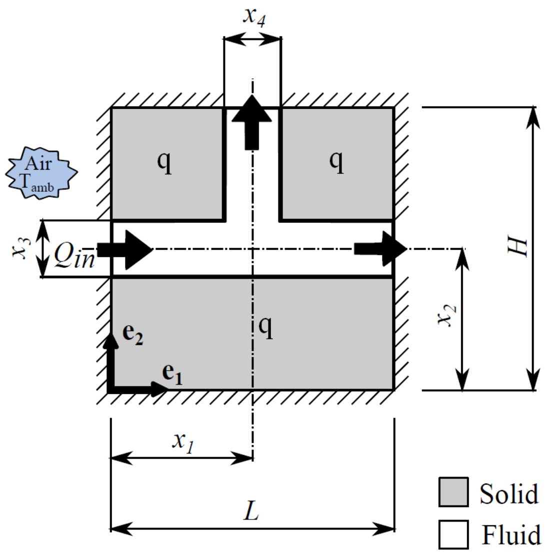
$-\mathrm{div}(\phi(x,y)) + q(x,y) = 0$, in the T-junction fluid and solid region **(3)**

$\phi(x,y) = 0$, on the solid boundaries B_s **(4)**

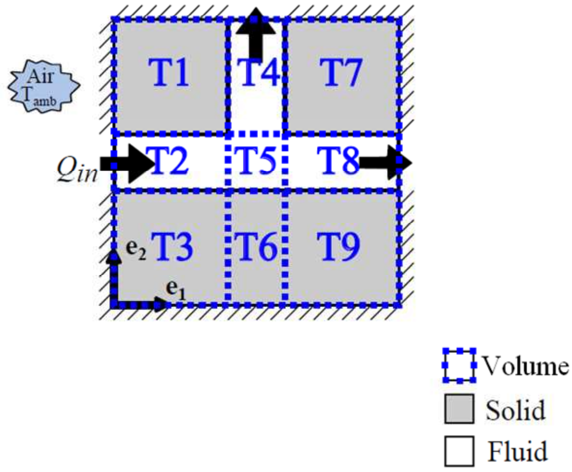
$\phi(x,y) = \rho c_p T_{Bf}$, on the fluid boundaries B_f **(5)**

in which,

$\phi(x,y)$ is the heat flow in $W.m^{-2}$
$q(x,y)$ is the heat term source in $W.m^{-3}$



(a) T-junction geometry



(b) Discrete domain

Figure 1: The T-junction domain

Equation (3) and the T-junction geometry are discretized using the lumped method. This discretization methodology is fully described in [7, 10, 16, 17, 19]. The discrete domain is presented in Figure (1.b), where one temperature value is assigned at each volume. Heat sources of 1000 *W* are placed in each solid volume. The resulting discretized domain is thermally represented by nine temperature values.

Considering a system with an available ventilation power of 4.6 *W*, global dimensions of (1 *m*,1 *m*) and ambient temperature of T_{amb} = 298 K, the objective is to optimize the cooling process of this system, by changing only the geometric variables *x*. For consistency, the fluid surface should remain the same ($A_{fluid} = 0.28\ m^2$), so that the temperature field is function of the variables x_1, x_2 and x_3, that is:

$$x_4 = (A_{fluid} - x_3 L)/(H - x_2 - 0.5x_3) \qquad (6)$$

$$T_i = f(x_1, x_2, x_3), \text{ for } i = 1, 2, \ldots, 9 \qquad (7)$$

From the nine temperature values, we can define two distinct criteria: the mean temperature, which takes into consideration the overall thermal behavior and the maximum temperature, a local thermal evaluation. The cooling improvement is measured by combining these two criteria.

3.2 Mathematical formulation of the problem

The T-junction problem is mathematically formulated as a constrained non-linear program.
The objective function *F* of the model is to minimize an aggregation of temperatures: the mean temperature T_{ave} and the maximum temperature T_{max}.

$$T_{ave} = \frac{\sum_{i=1}^{9} T_i}{9} \qquad (8)$$

$$T_{\max} = \max(T_i)_{i=1..9} \qquad (9)$$

F is given by this combination. Where $\alpha \in [0, 1]$.

$$F = \alpha * T_{max} + (1-\alpha)*T_{ave} \qquad (10)$$

The constraints of this problem are as follows:

$$x2 + 0,5x3 - 0,9H \leq 0 \qquad (11)$$

$$-x2 + 0,5x3 + 0,1H \leq 0 \qquad (12)$$

$$x_1 + 0,5\left(\frac{A_{fluid} - x_3 L}{H - x_2 - 0,5x_3}\right) - 0,9L \le 0 \tag{13}$$

$$-x_1 + 0,5\left(\frac{A_{fluid} - x_3 L}{H - x_2 - 0,5x_3}\right) + 0,1L \le 0 \tag{14}$$

The constraints (11) and (12) ensure that the horizontal channel will not to be too high or too low. The constraints (13) and (14) avoid that a position of the vertical channel will be too right or too left.

The variables $x = (x_1, x_2, x_3)$ are respectively bounded as follows

$$[0,1 \quad 0,1 \quad 0,02]^T \le x \le [0,9 \quad 0,7 \quad (A_{fluid} \text{-} 0,01)]^T$$

3.3 Solving the T-junction problem by PSO-TS

The PSO-TS method presented in section 2.2 is applied to the T-junction problem according to the design depicted in the programming flow chart of figure 2.

For this problem, the dimension of the swarm is three as the number of variables ($x = (x_1, x_2, x_3)$).

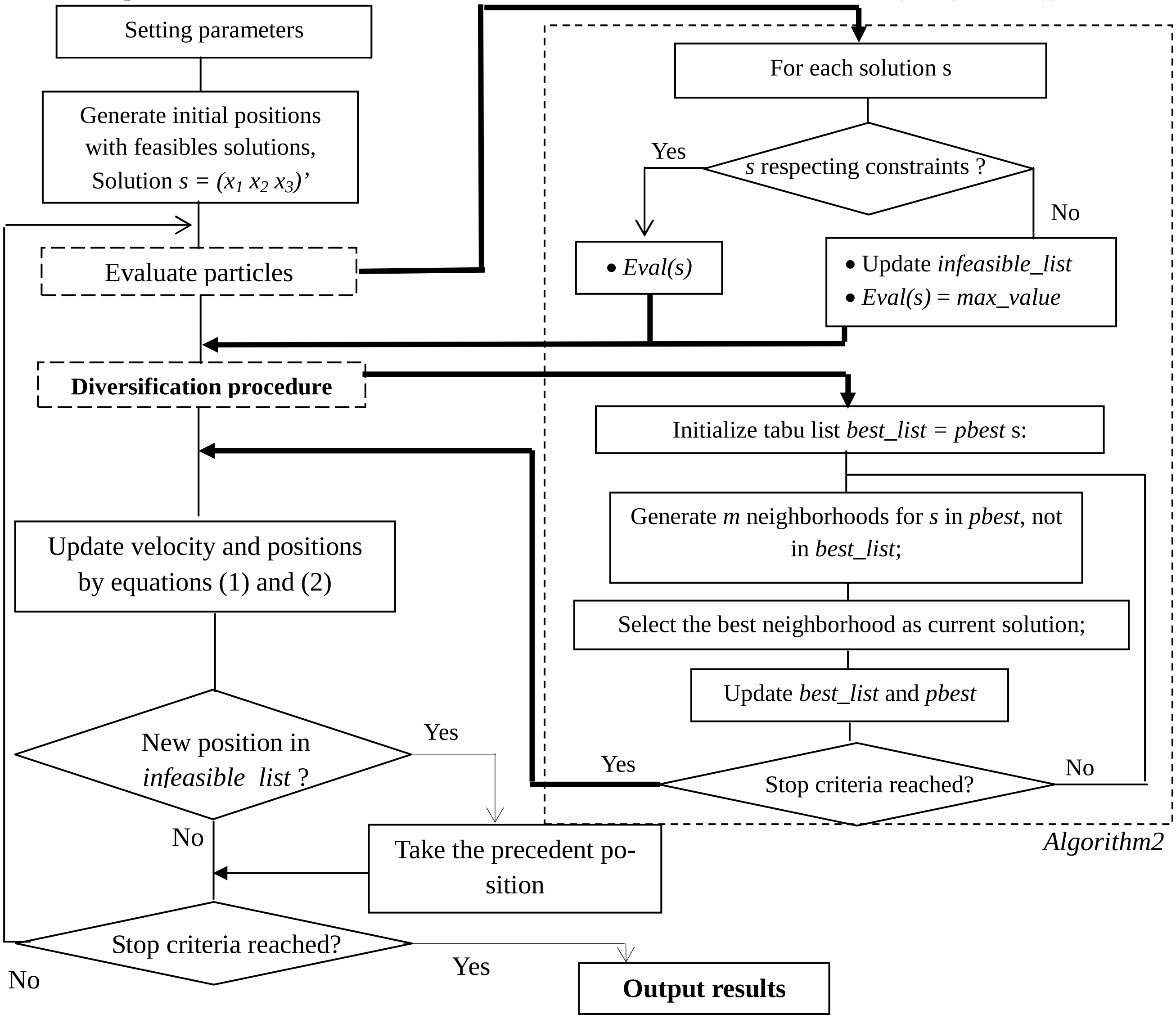


Figure 2: Programming flow chart of PSO-TS

3.4 Simulation results

Our PSO and PSO-TS are coded in Matlab 7.8 run on personnel computer (Intel Dual-Core processor/1.73GHz with 2GB of RAM). The two methods are compared to the genetic algorithm of the Matlab Toolbox.

To validate our methods, different configurations are tested on the T-junction problem. For each configuration, PSO, the genetic algorithm and PSO-TS are compared. The aims is to know which method is better in solution converging.

The parameters of PSO and PSO-TS are:

- Initial inertia weight *w* which is initialised at 0.9 and then decreased until 0.4,
- Acceleration coefficients c_1 and c_2 that both equal to 2,
- The size of *infeasible_list and best_list* are set to 10 and 6, for PSO-TS.

The parameters of the genetic algorithm are:

- The probability of mutation which has been set to 0.2 according to default configuration,
- The population size has been fixed to 20,
- The crossover fraction has been set to 0.8.

In the first tests, the number of generations is varied and we look to the convergence of the three methods.

The second tests concern PSO and PSO-TS. The swarm size is varied to compare the convergence of each method. In all tests we considered the objective as the average temperature (α = 1).

For all tests, 10 runs are carried out. The mean value (*Moy*), best and worst solutions (*Min and Max*) are saved. *Tm* represents the computation time. *NN* indicates that computing is not necessary.

#Generations	**GA**				**PSO**				**PSO-TS**			
	Moy	**Min**	**Max**	**Tm**	**Moy**	**Min**	**Max**	**Tm**	**Moy**	**Min**	**Max**	**Tm**
5	420,89	417,28	436,83	3665,10	420,75	416,22	424,46	186,83	416,53	415,59	417,42	598,13
10	416,69	415,64	420,05	7140,66	420,51	416,51	424,46	1331,46	415,80	415,60	416,07	958,87
20	416,28	415,54	417,05	7340,66	419,87	415,56	422,71	286,77	415,73	415,59	415,90	1940,79
30	415,81	415,54	416,10	11181,22	419,87	415,71	424,46	326,98	415,65	415,55	415,84	3185,55
60	**415,54**	**415,54**	**415,54**	**25765,26**	418,91	415,56	424,46	484,21	**415,57**	**415,55**	**415,60**	**8476,19**
1000	*NN*	*NN*	*NN*	*NN*	419,28	416,13	423,81	1061,51	415,56	415,54	415,58	14958,53
2000	*NN*	*NN*	*NN*	*NN*	419,92	418,06	421,46	1411,35	*NN*	*NN*	*NN*	*NN*

Table 1: comparison between GA, PSO and PSO-TS (variation of number of generations)

As shown in table 1, the genetic algorithm converged for 60 generations; however it took a great amount of time (more than 7 hours). This is due to the excessively large number of fitness function calls (5120 times for 60 generations). The PSO method converges slowly for this configuration, but as shown in the Table 2, it converges better than GA for larger sizes of the swarm. In both cases, the computation time of PSO is much smaller than that of the GA (on average running 27 times faster than the GA).

The PSO-TS outperforms both of these methods, beginning its convergence within 10 generations and taking much less time than the GA.

In Table 2, we present a comparison of PSO and PSO-TS by varying the swarm size and the number of generations. One can see that PSO-TS converges quickly whereas PSO needs more particles and generations to converge. This can be explained by the improvement of the solution diversity in PSO-TS brought about by the tabu search procedure. For example, PSO-TS converged with 75 particles and 50 generation and spent 1963, 55 seconds. However, PSO needed 300 particles and 100 generations to converge, and spent 3276.02 seconds.

Swarm size	#gene-ration	PSO				PSO-TS			
		Moy	Min	Max	Tm	Moy	Min	Max	Tm
50	10	418,90	416,69	421,54	504,38	415,89	415,62	416,58	1494,22
50	50	418,16	416,04	421,36	577,02	415,86	415,62	416,22	1537,62
75	50	418,28	415,78	420,49	925,17	**415,54**	**415,45**	**415,54**	**1963, 55**
75	100	417,58	415,78	419,53	989,24	*NN*	*NN*	*NN*	*NN*
75	150	417,47	415,78	419,92	1179,27	*NN*	*NN*	*NN*	*NN*
85	100	417,09	415,78	419,70	1279,15	*NN*	*NN*	*NN*	*NN*
100	100	417,37	415,78	419,70	1289,29	*NN*	*NN*	*NN*	*NN*
100	500	417,87	415,78	421,53	1429,73	*NN*	*NN*	*NN*	*NN*
150	100	416,59	415,67	418,03	1627,32	*NN*	*NN*	*NN*	*NN*
200	100	416,93	415,67	418,54	1966,30	*NN*	*NN*	*NN*	*NN*
250	100	416,42	415,67	417,24	2841,11	*NN*	*NN*	*NN*	*NN*
300	100	**416,45**	**415,66**	**417,51**	**3276,02**	*NN*	*NN*	*NN*	*NN*
150	300	416,76	415,60	418,22	2637,24	*NN*	*NN*	*NN*	*NN*
200	60	416,76	415,60	417,37	2051,79	*NN*	*NN*	*NN*	*NN*
250	60	416,58	415,60	417,55	2418,90	*NN*	*NN*	*NN*	*NN*
300	60	416,60	415,69	417,78	2843,58	*NN*	*NN*	*NN*	*NN*
500	60	416,39	415,86	417,27	4588,31	*NN*	*NN*	*NN*	*NN*
500	100	416,03	415,60	416,65	5131,66	*NN*	*NN*	*NN*	*NN*
500	200	416,23	415,84	416,71	4681,82	*NN*	*NN*	*NN*	*NN*

Table 2: comparison of PSO and PSO-TS (variation of swarm size and number of generations)

4 Conclusion and perspectives

In this paper we proposed a new algorithm devoted to the optimization of a non linear thermal problem, which led to a constrained non-linear optimization program (CNOP). To solve the CNOP, we used a particle swarm optimization method together with an embedded tabu search procedure which was used to improve the performance of the PSO. The new method, denoted PSO-TS, outperforms the basic PSO and a genetic algorithm in terms of solution quality and computation times. This improvement makes it possible to consider larger and more realistic case studies for the T junction problem, which was not possible before the use of the PSO-TS algorithm.

The PSO-TS can be further improved in two main ways. First, we have embedded only the simplest form of tabu search in our procedure and more advanced uses of its adaptive memory processes (see, for example, Glover and Laguna, 1997 [12]) are certain to yield gains. Second, to enhance our treatment of feasibility, we are studying the use of non-stationary penalty function and other techniques to handle constraints of the CNOP. This will enable us to improve the diversification procedure by making it more generic and applicable to other CNOP problems.

Looking forward, we additionally plan to compare our PSO-TS approach with other swarm optimization algorithms for thermal problems as reported in [17, 18].

Acknowledgement

The authors would like to thank the reviewer for his constructive and judicious comments. The authors are grateful to the financial support provided from the MEDEE project, the Région Nord Pas-de-Calais and JEUMONT Electric.

References

[1] Babu, B. and Angira, R., *Modified differential evolution (MDE) for optimization of non-linear chemical processes*', Computers & Chemical Engineering 30(6-7), 989 - 1002, (2006)

[2] Basri S., Kamarudin S., Daud W., Ahmad M., *Non-linear optimization of passive direct methanol fuel cell (DMFC)*, International Journal of Hydrogen Energy 35(4), 1759 - 1768 , (2010),

[3] Benli B. and Kodal S., *A non-linear model for farm optimization with adequate and limited water supplies: Application to the South-east Anatolian Project (GAP) Region*, Agricultural Water Management 62(3), 187 - 203(2003).

[4] Chelouah R. and Siarry P., *Enhanced continuous tabu search: An algorithm for the global optimization of multiminima functions*, in: S. Voss, S. Martello, I.H. Osman, C. Roucairol (Eds.), Metaheuristics: Advances and trends in local search paradigms for optimization, Kluwer, Dordrecht, 1999, pp. 49–62.

[5] Clerc M and Kennedy J, *The Particle Swarm. Explosion, stability, and convergence in a multidimensional complex space*, IEEE Transactions on Evolutionary Computation, 6 (2002) 58-73.

[6] Clerc, M., *Particle Swarm Optimization*, ISTE (2006),

[7] D'Arcier W. F. and Sérillon L., *Thermal Modelling of Permanent Magnet Motor for Traction*, Royal Institute of Technology, Master thesis, Stockholm, 2007

[8] Dong Y., Tang J., Xu B., Wang D., *An application of swarm optimization to nonlinear programming*, Journal Computers & Mathematics with Applications, Volume 49 Issue 11-12, June, 2005

[9] Eberhart R.C. and Kennedy. J, *A New optimizer using particle swarm theory, Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995.

[10] EL-Refaie A.M., Harris N.C., Jahns T.M., Rahman K.M., *Thermal Analysis of Multibarrier Interior PM Synchronous Machine Using Lumped Parameter Model*. IEEE Transactions on, Energy Conversion. Volume 19, Issue 2, pages: 303-309, June 2004

[11] Gamot R. M., Mesa A., *Particle swarm optimization: Tabu search approach to constrained engineering optimization problems*, WSEAS Transactions on Mathematics, Volume 7 Issue 11, November 2008

[12] Glover F. and Laguna M., *Tabu Search*. Kluwer, Boston, 1997.

[13] Hu X. and Eberhart R.C, *Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization*, Proceedings of the Sixth World Multi-conference on Systemics, Cybernetics and Informatics 2002 (SCI 2002), Orlando, USA. 2002,

[14] Kennedy. J and Eberhart R.C, *Particle Swarm Optimization*, IEEE International Conference on Neural Networks, Perth, Australia, 1995.

[15] Leyffer S. , Mahajan A., Cochran J. J., Cox, L. A., Keskinocak P., Kharoufeh J. P., Smith J. C., *Software For Nonlinearly Constrained Optimization*, John Wiley & Sons, Inc. 2010.

[16] Lindstrom L., *Thermal Model of a Permanent-Magnet Motor for a Hybrid Electric Vehicle*, Chalmers University of Technology, Internal report, Göteborg, Sweden, 1999

[17] Patel V. and Rao R., *'Design optimization of shell-and-tube heat exchanger using particle swarm optimization technique*', Applied Thermal Engineering 30(11-12), 1417 – 1425, 2010.

[18] Rao R. and Patel V. (), *'Thermodynamic optimization of cross flow plate-fin heat exchanger using a particle swarm optimization algorithm'*, International Journal of Thermal Sciences 49(9), 1712 – 1721, 2010.

[19] Rouhani H., Faizb J., Lucas C., *Lumped thermal model for switched reluctance motor applied to mechanical design optimization*, Mathematical and Computer Modelling, vol. 45, pp. 625–638, 2007

[20] Sha D.Y., Hsu C.Y, *A hybrid particle swarm optimization for job shop scheduling problem*, Computers & Industrial Engineering 51 (2006) 791–808.

[21] Shelokar P. S., Siarry P, Jayaraman V. K, Kulkarni B. D, *Particle swarm and ant colony algorithms hybridized for improved continuous optimization*, Applied Mathematics and Computation 188 (2007) 129–142.

[22] Shen Q, Shi W-M, Kong W, *Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data*, Computational Biology and Chemistry 32 (2007) 53–60.

[23] Shi Y.H., and Eberhart R.C, *A modified particle swarm optimizer*, in: Proc. of the IEEE Congress on Evolutionary Computation. IEEE Service Center, USA (1998) 69–73.

[24] Thangaraj R., Pant M., Ajith Abraham A., Pascal Bouvry P., *Particle swarm optimization: Hybridization perspectives and experimental illustrations*, Appl. Math. Comput. (2011), doi:10.1016/j.amc.2010.12.053

[25] Trelea I.C, *The particle swarm optimization algorithm: convergence analysis and parameter selection*, Information Processing Letters, 85 (2003) 317-325.