Tabu Search for the Minimum Number of Matches Problem (A Heat Exchanger Network sub-problem)

Outline of the method

TS is a global approach for chiefly solving combinatorial optimization problems; the approach uses a neighborhood search and incorporates an adaptive memory for a responsive exploration to evaluate multiple regions of the solution space, allowing escaping from a local optima solution. TS has been proved to be a useful tool for a wide range of problems in different research areas, ranging from the classic operation research such as scheduling problems and the traveling salesman problem, to engineering applied problems such as heat exchanger network synthesis, pump system configuration and a Wastewater Treatment Plant.

The minimum number of matches is a sub-problem of the original heat exchanger network synthesis (HENS) problem and represent the major bottleneck in sequential methodologies.

Usually the sequential methodologies decompose the HENS problem in three specifies targets resolved hierarchically according to the impact they have on the cost of the final network design. The classic tasks to solve are. i) Minimum utility usage formulated as a linear programming (LP) model, ii) Minimum number of matches formulated as a mixed integer linear programming (MILP) model, and iii) The investment cost as a nonlinear (NLP) programming model.

Even when the sequential methodologies reduce the computational complexity of the HENS problem, the minimum number of matches sub-problem is characterized as a *NP-Hard* in the strong sense problem. Here, a TS strategy is implanted for solving the minimum number of matches problem in heat exchanger networks as a first approach of a meta-heuristic technique for trying to overcome the actual limitations of this problem.

The strategy used in this work consists of using TS only for fixing de discrete variables, isolating the remaining constraints and formulating them as a system of linear equations, which can be solved with good calculation speed using, for instance, commercial Linear Programming (LP) solvers.

Neighborhood Selection

At each iteration, a collection of neighbor solutions will be generated starting from a modification of the current one, the process is described next.

1. A set of binary variables y_{ij} is randomly generated to determine the possible matches between hot and cold streams.
2. With the fixed variables from Step 1, a modified version of the minimum matches sub-problem with no hot/cold utilities is solved as a linear programming problem, to obtain the heat duties q_{iljl1}.
3. If the problem results infeasible, return to Step 1.
4. The heat duty of hot and cold utilities, $qHU_{j,l1}$, $qCU_{i,l}$, are determined according to the overall balance of each stream.
5. If utilities are required for a stream, indicates the existence of a match between a process stream setting the binary variables yHU_j, yCU_i.

At each TS iteration, the neighborhood of the current solution is generated, and one neighbor is selected as a starting point for the next iteration. The move operator to generate the neighbor solutions consists of applying a random change in a single binary variable 0/1. This means that some match could be add or remove from the current solution.

The number of evaluated solutions of TS correspond to the product of the neighbors generated (N_{Neigh}) and the total of the iterations number M. Lin (2002) reports a heuristic for setting this number as follows.

$$N_{Neigh} = \kappa N_{\mathrm{var}}^2$$

Where κ is a free parameter which determines the number of neighbors generated at each iteration (recommended values are 1 and 10) and N_{var} is the number of continuous variables. Lin (2002) found that setting M to either 100 or 500 works with a good balance between an efficient computational time and consistent improvement for new quality solutions.

Theoretically, an infinite number of neighbors would ensure the global optimality of the TS, however, in the practice this is impossible to implement due to the required computational resources. As good solutions wish to be determined in reasonable time, good neighbor generation plays an important role along the search.

Intensification Strategy.

In TS, intensification is a widely used strategy which opens the possibility of focus in some promising areas more deeply; this can enhance the performance of the global search by exploring the neighbors of an elite group of solutions (Glover and Laguna, 1997). This statement can be achieved, for instance, by maintaining the same structure fixing the binary variables and only changing values of the continuous variables to obtain a lower objective function values. Nevertheless, in the minimum number of matches sub-problem for heat exchanger networks, the value of the objective function does not depend of the continuous variables, which only needs to be checked for feasibility in the linear constraints. With this insight of the problem, the intensification strategy can be removed, since the solutions generated by TS only needs to be feasible.

Diversification.

Diversification is another strategy to improve the performance of TS, allowing explore certain areas that has not been frequently investigated, this can be done by forbidding some moves that involves solutions or areas already visited, and records them into lists (Tabu lists). As already been settled, the minimum number of matches sub-problem only needs to check feasibility for a generated solution, so in this case, the Tabu lists only keeps track of the binary variables. The diversification approach makes use of two list of short and long term memory named recency-based and frequency-based respectively. The recency-based list, classifies as Tabu the neighbors that are not better than the current solution, these solutions remain on the list throughout the Tabu tenure, which is the property that indicates the time of a solution is forbidden. The solutions from the bottom of the list are released when new solutions are classified Tabu, this short-term memory prevent infinite cycling, avoiding getting easily trapped in local optima.

The frequency-based Tabu list provides the long-term memory which forbids solutions that are been frequently visited, this long term-memory saves information from the overall search process, to evaluate different regions of the search space. The tracking is done by adding indexes; when a solution is revisited its index is increased and otherwise the index decreases. The solutions in the frequency-based Tabu list can be released if their index is lower than a threshold, in the opposite case, when a solution index is higher than a predefined threshold (which is equal to the length of the frequency-based tabu list) it means that TS has been looking in certain area for a long period, so the process is forced to select a new random solution restarting the search; this avoids that the search get trapped in local optima.

The memory-based framework of TS gives it some advantages among other meta-heuristics memoryless methodologies (e.g. Simulated annealing, Genetic Algorithms), since it provides an informed search with continuously updated information, increasing the probabilities to obtain attractive solutions.

Aspiration criterion

In addition, an aspiration criterion can be implemented to override the Tabu property in order to increase the generation of more diversified solutions towards the end of the TS. The approach is done by comparing, at each iteration, some random number (0 <= P <= 1) and a function that increases as the search progresses. If the P is greater than the function, the aspiration criterion is satisfied, so the tabu property remains, and the best non-neighbor is used as a starting point for the next iteration, this encourage a more diverse search. When the aspiration criterion is not satisfied, the tabu property is overridden cancelling the restart process resulting from a frequency-based Tabu list and allowing Tabu solutions from the recency-based Tabu list to generate neighbors from the next iteration. As the search go forward, the probabilities of override the tabu property increases, allowing a more intense search of a given area.

The minimum number of matches sub-problem does not require an intensified search and override the tabu property would mean a step back in the search because there is no need of revisit old solutions as they cannot improve the objective function. For this reason, the aspiration criterion is discarded.

Minimum Number of Matches Problem – MILP

$$\text{Min} \quad \sum_{i\in I}\sum_{j\in J} y_{ij} + \sum_{j\in J} yhu_j + \sum_{i\in I} ycu_i \tag{1}$$

Subject to:

Supply and demand constraint

$$s\sum_{j\in J}\sum_{l=k}^{\text{NOK}} q_{i,k,j,l} + qcu_{i,k} = \text{HS}_{i,k} \qquad i\in I;\ K=1,\ ...,\ \text{NOK} \tag{2}$$

$$\sum_{i\in I}\sum_{k=1}^{l} q_{i,k,j,l} + qhu_{j,l} = \text{HD}_{j,l} \qquad j\in J,\ \ m=1,\ ...,\ \text{NOK} \tag{3}$$

Logical constraint

$$\sum_{k=1}^{\text{NOK}}\sum_{l=k}^{\text{NOK}} q_{i,k,j,l} \le \text{M}_{ij}\cdot y_{ij} \qquad i\in I,\ \ j\in J \tag{4}$$

$$\sum_{l=1}^{\text{NOK}} qhu_{jl} \le \text{Mhu}_j \cdot yhu_j \qquad j\in J \tag{5}$$

$$\sum_{k=1}^{\text{NOK}} qcu_{i,k} \le \text{Mcu}_i \cdot ycu_i \qquad i\in I \tag{6}$$

Where $\text{M}_{i,j}$, Mhu_j, Mcu_i is the maximum amount of heat that can be exchanged between two process streams or utilities.

Non-negativity constraints

$$q_{i,k,j,l},\ qhu_{j,l},\ qcu_{i,k} \ge 0 \qquad \forall i\in I,\ j\in J,\ k\in IK_i,\ l\in JL_j \tag{7}$$

$$y_{i,j}, yhu_j, ycu_i = 0{,}1 \qquad \forall i\in I,\ j\in J \tag{8}$$

Fig. 1 shows the schematic diagram of the particular TS algorithm used to solve the minimum matches problem.

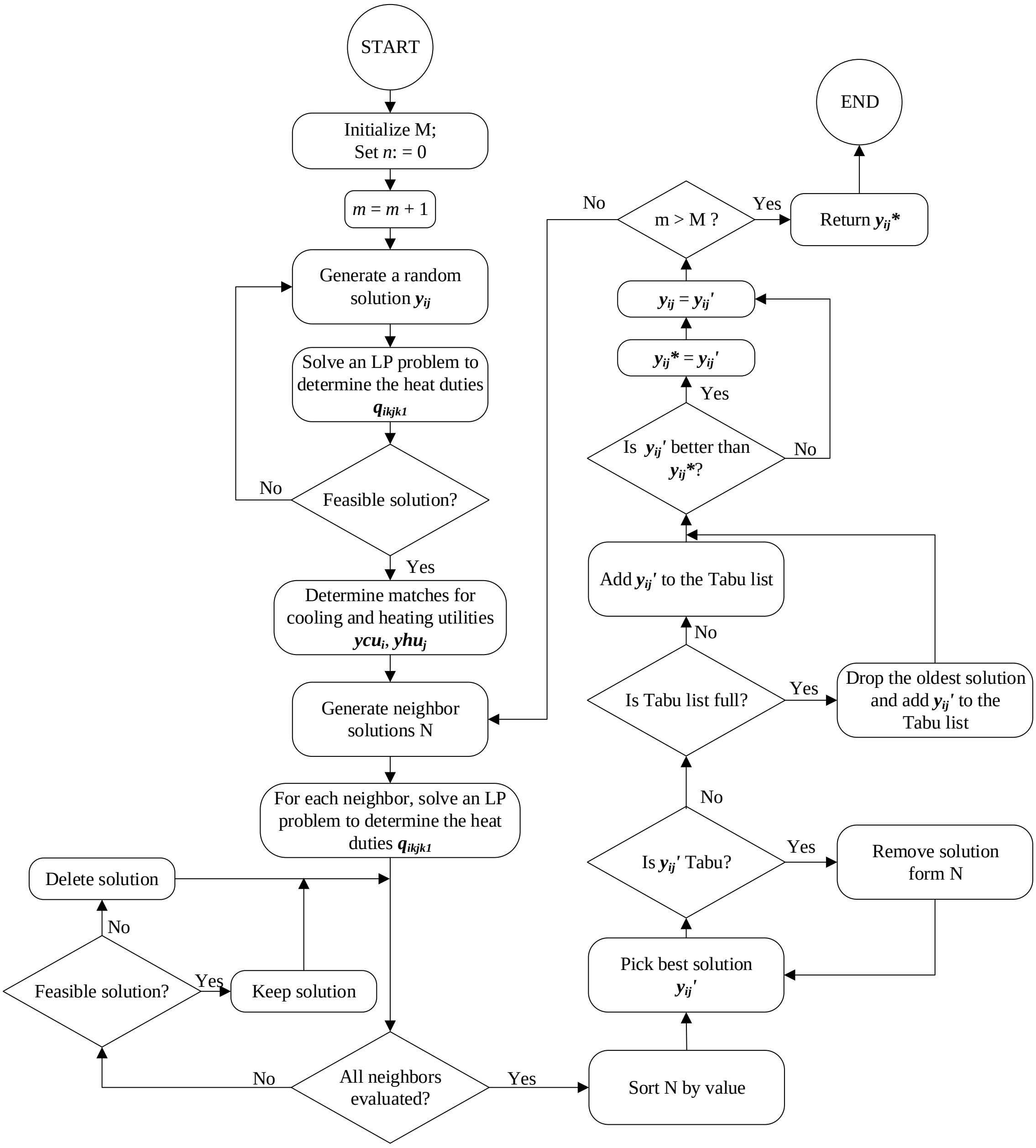

Fig. 1. Tabu Search algorithm for the minimum number of matches problem.

Results

All calculations were performed in a personal DELL XPS 8500 CPU Intel i7-3770 3.7 GHz with 16 GB of RAM memory running windows. An interface was made between C to program the TS algorithm and GAMS/CEPLEX to solve the LP models. Ten trials were performed for each case of study.

Case 1. The HEN test problem was addressed in the context of the minimum number of matches sub-problem. The case is the classic 10SP1 problem, involves five hot/ five cold process streams, one hot and one cold utility. The HRAT was selected with a value of 10 K. 18 temperature intervals were generated. Stream data are given in Table 1. The problem has a total of 30 binary variables (5 define matches between hot stream process and cold utilities, no hot utilities are required). In addition, there are 8316 unknown continuous variables defining heat exchanges. During the neighbor generation process, the heat duties are determined through the relaxed LP model without binary variables. The binary variables corresponding to stream/stream heat exchange are selected randomly and keep in track using the TS algorithm. The binary variables of heat/cooling utilities are determined to meet the required outlet temperatures. The default number for κ is 2, using $M = 20$ the global optima for this case were determined for 8 out of 10 trials.

Case 2. The example is a large-size problem known as 20SPI. The problem involves ten hot/ ten cold process streams, one hot and one cold utility. The HRAT was selected with a value of 10 °F. 32 temperature intervals were generated. Stream data for this problem are detailed in Table 2. With the transportation formulation a total of 110 binary variables and 103161 continuous variables are generated. The value of κ is 1 and using $M = 50$ the best objective value obtained is 20 in all the tests cases. The sub-optimal solution may be caused because the LP models solved to get the heat loads distribution of the transportation model, are only oriented to fulfil the total of energy constraints, but no control of the number of the number of utilities heat exchangers is take into account.

Table 3 summarizes the size of the problems and the number of variables required for each one. In table 4 are presented the results of the Tabu search, compared with solutions and times produced by a deterministic MILP solver (CPLEX).

Table 1. Stream data for Case of Study 10SP1

Corriente	Temperatura de suministro T(K)	Temperatura de objetivo T(K)	Flujo de capacidad calorífica F(kW K^{-1})
H1	433	366	8.79
H2	522	411	10.55
H3	544	422	12.56
H4	500	339	14.77
H5	472	339	17.73
C1	355	450	17.28
C2	366	478	13.90
C3	311	494	8.44
C4	333	433	7.62
C5	389	495	6.08
HU	509	509	—
CU	311	355	—

Table 1. Stream data for Case of Study 10SP1

Corriente	Temperatura de suministro T(°F)	Temperatura de objetivo T(°F)	Flujo de capacidad calorífica F(Btu h^{-1} $°F^{-1}$)
H1	550	450	28,400
H2	520	480	23,870
H3	500	440	24,600
H4	460	370	17,000
H5	415	340	30,690
H6	400	300	19,360
H7	365	320	25,500
H8	300	250	12,420
H9	240	170	20.72
H10	170	140	18,900
C1	375	420	22,320
C2	300	370	23,870
C3	320	360	34,960
C4	280	340	26,040
C5	260	330	13,440
C6	225	280	32,000
C7	240	265	11,100
C8	170	220	22,960
C9	140	200	14,400
C10	80	140	13,920
HU	456	456	—
CU	100	180	—

Table 3. Problem sizes of the test cases. The number of variables and constraints are computed with respect to the transportation model.

Case of Study	Hot Streams	Cold Streams	Temperature intervals	Binary variables	Continuous variables	Constraints
10SP1	5	6	18	30	8316	218
20SP1	10	11	32	110	103161	763

Table 4. Solutions to Case 1 and Case two TS vs deterministic solver, based on a transportation problem formulation.

	CPLEX MILP			Tabu Search		
Case of Study	Value	CPU Time (s)	Gap	Best solution	Iterations	Average CPU Time (s)
10SP1	10	2.921	-	10	20	0.852
20SP1	19*	-	29%	19	50	508.490

* Time limit of 30 min was reached and no exact solution is guaranteed.