

doi: 10.3969/j.issn.1006-1576.2010.04.003

基于改进蚁群算法的战时雷达器材运输调度模型

彭飞¹, 杨江平², 钱建刚³, 项建涛²

(1. 空军雷达学院 研究生管理大队, 湖北 武汉 430019; 2. 空军雷达学院 陆基预警监视装备系, 湖北 武汉 430019;
3. 空军雷达学院 预警监视指挥系, 湖北 武汉 430019)

摘要: 针对战时雷达器材运输中的有硬时间窗要求的运输调度问题, 提出了一种改进的蚁群算法。通过分析其模型的建立, 引入最大最小信息素约束, 用奖惩因子用于信息素的更新, 局部搜索策略 2-opt 和 or-opt。并通过实例进行验证, 结果表明, 在雷达器材需求点数目和需求数量等各种条件已知的情况下, 该算法能得到较好的全局最优解, 比基本蚁群算法具有更快的收敛速度和更高的收敛精度, 并对其它相关的运输调度问题有很强的借鉴意义。

关键词: 硬时间窗; 战时; 雷达器材; 运输调度

中图分类号: TP301.6; C934 **文献标识码:** A

Model of Wartime Radar Equipment Transportation Routing Based on Improved Ant Colony Algorithm

PENG Fei¹, YANG Jiang-ping², QIAN Jian-gang³, XIANG Jian-tao²

(1. Administrant Brigade of Postgraduate, Air Force Radar Academy, Wuhan 430019, China;
2. Dept. of Land-Based Early Warning & Surveillance Equipment, Air Force Radar Academy, Wuhan 430019, China;
3. Dept. of Early Warning & Surveillance Command, Air Force Radar Academy, Wuhan 430019, China)

Abstract: Aiming at the radar equipment transportation problem with hard time windows in wartime, an improved ant colony algorithm is presented. The improved algorithm adopted Max-Min information element restriction, and adopted rewards and punishment genes to update information element, and adopted local searching strategy 2-opt and or-opt. Given the radar equipment customer number and demand quantity and all the condition, the algorithm can obtain the preferable global solving result. The improved algorithm has faster convergence rate and has higher accuracy, and also has instructional meaning to others transportation problem.

Keywords: Hard time window; Wartime; Radar equipment; Transportation routing

0 引言

战时雷达器材调度属于硬时间窗问题, 战时每个雷达器材需求点需要补充雷达器材的时间不同, 且雷达器材必须在要求的时间段内补充。带时间窗的车辆路径问题已经被证明为 NP-hard 难题, 启发式算法成为解决该问题的有效途径, 90 年代初, 蚁群算法的出现为解决带时间窗的车辆路径问题提供了新的工具, 但传统蚁群算法仍然存在着运算时间长、容易陷入局部优化等问题。故提出一种改进的蚁群算法来求解带硬时间窗的运输车辆调度问题。

1 模型建立

战时某雷达器材供应点给多个雷达器材需求点运送雷达器材, 已知各雷达器材的需求量和需求时间段, 雷达器材供应点与雷达器材需求点、雷达器材需求点与雷达器材需求点两两间的运输时间、运输距离, 及运输车辆在各需求点的服务时间等条件, 器材供应点计划用多台运输车辆从雷达器材供应点

出发, 分别按顺序到达各雷达器材需求点, 当累积载货量超出单车容量限制或到达时间超出时间窗时, 则该车辆完成运输任务返回供应点。要求为每辆雷达器材运输车辆确定行驶线路, 在满足各雷达器材需求点的硬时间窗约束和车辆容量约束的条件下, 使运输距离总和最短。基于硬时间窗约束的战时雷达器材运输调度问题的数学模型为:

目标函数:

$$\min \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^K E_{ij} x_{ijk} \quad (1)$$

约束条件:

$$\sum_{j=1}^n x_{0,jk} = 1, k \in K \quad (2)$$

$$\sum_{i=1}^n x_{i,0k} = 1, k \in K \quad (3)$$

$$\sum_{k=1}^K \sum_{j=0}^n x_{ijk} = 1, i \in \{1, 2 \dots n\} \quad (4)$$

收稿日期: 2009-10-16; 修回日期: 2009-12-14
基金项目: 雷达装备战场抢修能力建设资助项目
作者简介: 彭飞 (1981-), 男, 江西人, 博士生, 从事装备保障研究。

$$\sum_{k=1}^K \sum_{i=0}^n x_{ijk} = 1, j \in \{1, 2, \dots, n\} \tag{5}$$

$$\sum_{i=0}^n D_i \sum_{j=0}^n x_{ijk} \leq Q \tag{6}$$

$$\sum_{k=1}^n \sum_{i=0}^n x_{ijk} (T_{ij} + T_i + F_i) = T_j, j \in \{1, 2, \dots, n\} \tag{7}$$

$$ET_i \leq T_i \leq LT_i, i \in n \tag{8}$$

$$T_0 = F_i = 0 \tag{9}$$

$$x_{ijk} = \begin{cases} 1 & \text{车辆 } k \text{ 从需求点到需求点 } j \text{ 时, } i, j \in \{1, 2, \dots, n\} \\ 0 & \text{其它} \end{cases}$$

式中， n 为需求点总数， k 为运输车辆总数， E_{ij} 为需求点 i 与需求点 j 之间的距离， T_{ij} 为从需求点 i 到需求点 j 所需的时间， D_i 为需求点 i 的需求量， F_i 为运输车辆在需求点 i 的服务（装卸搬运）时间， Q 为运输车辆的容量， T_i 表示运输车辆到达需求点 i 的时刻， ET_i 表示需求点 i 最早允许开始服务的时刻， LT_i 表示需求点 i 最晚必须完成服务的时刻。为方便研究，假设所有车辆容量相同，运输车辆在各需求点服务时间 F_i 为零。

目标函数表示运输距离总和最短，约束条件表示运输车辆从供应点出发，式 (3) 表示运输车辆完成运输任务后回到供应点，式 (4)、式 (5) 表示每个需求点雷达器材只由 1 辆运输车辆服务且所有需求点都得到服务，式 (6) 表示运输车辆 k 装载的雷达器材总量不超过该车辆的装载容量，式 (7) 表示运输车辆到达需求点 j 的时刻为运输车辆到达需求点 i 的时刻、在需求点 i 的服务时间和通过路段 (i, j) 的运输时间的总和，式 (8) 表示运输车辆必须在规定的时间内到达需求点，式 (9) 表示运输车辆出发时刻为零，在各需求点服务时间为零。

2 算法设计^[1-8]

蚁群算法具有正反馈、分布式计算以及贪婪的启发式搜索等主要特点，正反馈过程使得该算法能发现较好解；分布式计算使得该算法易于并行实现，更快地得到较好解，这些特点使蚁群算法在求解带时间窗的车辆路径问题时更具有优势。但传统蚁群算法仍然存在着容易陷入局部优化等问题，故提出了一种改进的蚁群算法来求解带时间窗的车辆路径问题。改进后的蚁群算法步骤如下：

1) 初始化各控制参数，迭代计数器 $NC=0$ ，设置 NC 最大值和各边初始信息素。

(1) 蚂蚁数目的初始化

蚂蚁数目越多，蚂蚁在一次迭代中找到更好解

的可能性就越大，但另一方面意味着一次迭代中的路径搜索计算越多，导致整个算法速度变慢，故需要对蚂蚁数目进行平衡，根据实验分析依据城市规模/蚂蚁数目=1.5 的比例来确定蚂蚁数目比较合适。

(2) 信息素挥发因子 ρ 初始化

随着时间的推移，路径上留下的信息会逐渐消失，当要处理的问题规模较大时，会使那些从来未被搜索的路径上的信息量减少到接近于 0，因而降低了算法的全局搜索能力，而且当 ρ 过大时，以前搜索过的路径再次被选择的可能性过大也会影响算法的随机性能和全局搜索能力；反之， ρ 过小虽然可以提高算法的随机性能和全局搜索能力，但又会使算法的收敛速度降低。由以往实验分析可知： $\rho=0.5$ 时算法的全局收敛性和收敛速度都很好，计算性能也比较稳定。

(3) 信息素的初始化

传统的蚁群算法容易出现搜索面的局部停滞，为避免这种早熟现象，将信息素水平限制在 $[\tau_{\min}, \tau_{\max}]$ ，并在开始搜索前将所有边的信息素水平设置为 τ_{\max} ，从而使蚂蚁在搜索初期具有较大的搜索范围。由此可见， τ_{\min} 、 τ_{\max} 的设定至关重要。采用动态策略来确定 $\tau_{\min}(t)$ 、 $\tau_{\max}(t)$ 。

最初信息素还未更新时有：

$$\tau_{\max}(t) = \frac{1}{2(1-\rho)} \cdot \frac{1}{L_{Global}} \tag{10}$$

信息素更新后有：

$$\tau_{\max}(t) = \frac{1}{2(1-\rho)} \cdot \frac{1}{L_{Global}} + \frac{\sigma}{L_{Global}} \tag{11}$$

$$\tau_{\min}(t) = \frac{\tau_{\max}(t)}{20} \tag{12}$$

σ 表示路径构造中排名前几位的蚂蚁数目。

(4) 启发式因子 α 初始化

反映蚂蚁在运动中所积累的信息量在指导蚁群搜索中的相对重要程度，其值越大，蚂蚁选择以前走过路径的可能性越大，搜索的随机性减弱，当 α 较小时则易使蚁群的搜索过早限于局部最优。经实验分析， $\alpha \in [1, 4]$ 时蚁群算法的综合求解性能较好。

(5) 期望启发式因子 β 初始化

期望启发式因子反映了启发式信息在指导蚁群搜索过程中的相对重要程度，其大小反映了蚁群寻优过程中先验性、确定性因素的作用强度。其值越大，蚂蚁在某个局部点上选择局部最短路径的可能性越大，但是蚁群搜索最优路径的随机性减弱，容易陷入局部最优。经实验分析， $\beta \in [3, 5]$ 时蚁群算法

的综合求解性能较好。

2) 按初始解算法找到全局初始解 L_{Global} 。

全局初始最优解时算法如下: 蚂蚁从客户中心出发根据客户选择策略逐个访问客户, 当该蚂蚁到达客户时间超出时间窗要求或者无法满足载重量要求时, 该蚂蚁返回客户中心并重新从客户中心出发按客户选择策略重新访问未被访问过的客户, 直至所有客户都被该蚂蚁访问过一次为止。客户选择策略是从当前客户出发, 选择所有未被访问的客户中开始服务时间最早且满足载重量约束的那个客户。

3) 建立各蚂蚁的访问表 $allowed(k)$ 。

$$P_{ij}^k = \begin{cases} \lambda_1 \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{p \in allowed(k)} (\tau_{ip})^\alpha (\eta_{ip})^\beta} + \lambda_2 \frac{1/(|T_j - ET_j| + |T_j - LT_j|)}{\sum_{p \in allowed(k)} 1/(|T_p - ET_p| + |T_p - LT_p|)}, & j \in allowed(k) \\ 0, & \text{其他} \end{cases} \quad (13)$$

其中, η_{ij} 为两点间距离 l_{ij} 的倒数; $[ET_j, LT_j]$ 表示客户 j 的时间窗, ET_j 是允许最早到达时间, LT_j 是允许最迟到达时间; λ_1 、 λ_2 为权重系数, $\lambda_1 \in [0.5, 1]$, $\lambda_2 \in [0, 0.5]$, 且 $\lambda_1 + \lambda_2 = 1$ 。

5) 清空访问表 $allowed(k)$, 以 j 点为起点, 将符合时间窗要求和载重量约束的点重新放入访问表 $allowed(k)$ 中, 若 $allowed(k)$ 为空则转到 6), 否则转到 4)。

传统的蚁群算法中, 当蚂蚁从客户 i 选择下一个客户 j 时, 是先根据转移概率公式计算所有未走过节点的转移概率, 再依据时间窗和载重量约束来判断该点是否可行, 这种方法会增加运算次数和时间, 为提高运算效率, 先将满足时间窗要求和载重量约束的点先放入访问表中, 再在这些点中根据转移概率选择下一节点。

6) 若 m 只蚂蚁完成搜索回到客户中心, 即每只蚂蚁访问了所有节点并且所有节点上蚂蚁数量为 0, 每只蚂蚁将得到若干条由客户中心为起点的回路, 每条回路相当于 1 辆车所经过的路径, 分别计算 m 只蚂蚁走过的路径长度并得出最优路径 L_{Local} 。

7) 对最优路径 L_{Local} 分别实施 n 次 2-opt 和 or-opt 优化 (n 初值取大, 随着代数的递增而递减), 每次交换后若该路径满足车辆载重量与时间窗约束, 则记录本次路径长度 L_{opt} 与对应的路径表, 最终 $L_{Local} = L_{opt}$ 。

① 2-opt

2-opt 邻域搜索优化算法能有效消除边的交叉现象, 设 $routeI$ 、 $routeJ$ 为本次迭代最优解中任意 2

为增加发现可行解的机会, 在每次迭代前, 将蚂蚁随机均匀分布于各个节点, 即将 m 只蚂蚁随机均匀放置在 m 个客户点上, 并分别为各蚂蚁建立访问表 $allowed(k)$, $k=1, 2, \dots, m$ 。

4) 对于蚂蚁 k , 对访问表 $allowed(k)$ 中的点依据转移概率公式来选择蚂蚁的下一个节点 j 。

每只蚂蚁在选择下一个客户时, 除满足时间窗要求和载重量约束条件, 还需要考虑通往下一城市的路径长度和和路径上的信息量, 以及优先考虑等待时间较短和时间窗较小的需求点, 因此蚂蚁 k 由客户 i 转向客户 j 的转移概率为:

条路径, 将 $routeI$ 中的节点 $(i, i+1, i+2, \dots)$ 与 $routeJ$ 中的节点 $(j, j+1, j+2, \dots)$ 进行交换, 若能改善目标函数, 则执行交换操作。

在图 1 中, 如果 $L_{i,j+1} + L_{j,i+1} < L_{i,i+1} + L_{j,j+1}$ 成立且满足时间窗要求和载重量约束条件, 则删除路段 $(i, i+1)$ 和 $(j, j+1)$, 同时增加路段 $(i, j+1)$ 和 $(j, i+1)$ 。通常的 2-opt 算法从某个起始点开始, 依次对 2 条边进行上述判断和操作。对一个顶点处理完之后, 再对下一个顶点进行同样的处理。

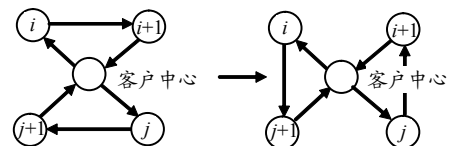


图 1 2-opt 算子原理

② or-opt

or-opt 主要用于路线内的改善, 该优化算法以单点 i 为主, 也有可能是连续 2 点 $(i, i+1)$ 或连续 3 点 $(i, i+1, i+2)$ 。找出该路径中客户 i 的 h 个邻近客户点, 并判断插入客户 j 与 $j+1$ 间是否改善目标函数, 若是, 则将客户点 i 插入到客户点 j 与 $j+1$ 之间。

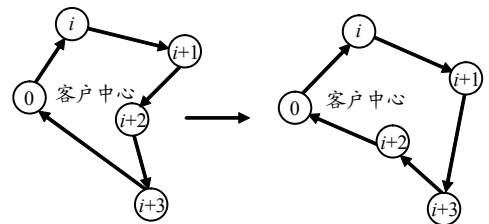


图 2 or-opt 算子原理

在图 2 中, 如果 $L_{i+1,i+3} + L_{i+2,0} < L_{i+1,i+2} + L_{i+3,0}$ 成立且满足时间窗约束条件, 则删除路段 $(i+1, i+2)$

和 (i+3, 0), 同时增加路段 (i+1, i+3) 和 (i+2, 0)。

8) 比较本次最优路径 L_{Local} 与全局最优路径 L_{Global} , 若 $L_{Local} < L_{Global}$, 则将本次最有路径作为全局最优路径 $L_{Global} = L_{Local}$ 。

9) 对最优路径上的信息素根据式 (11) 进行全局更新。

$$\tau_{ij}^{new} = (1 - \rho)\tau_{ij}^{old} + \sum_{k=1}^m \tau_{ij}^k + \frac{1}{L_{Local}} \quad (14)$$

$$\tau_{ij}^k = \begin{cases} \frac{X}{L_k} & \text{第 } k \text{ 只蚂蚁经过路段 } (i, j) \text{ 时} \\ 0 & \text{否则} \end{cases} \quad (15)$$

其中, X 是常数, 为蚂蚁每次施放的信息素, L_k 为第 k 只蚂蚁经过的路径长度。在蚁群算法的迭代早期, 各路径信息素差别不大, 蚂蚁主要依靠启发式信息寻找最优路径, 此时算法近似于贪心算法。算法迭代后期, 各路径上的信息素差别较大, 此时蚂蚁主要依靠信息素寻找路径, 由于蚁群算法具有正反馈的特点, 无论搜索到的解如何, 在这些被搜索到的路径上的信息素都会得到一定程度的增强, 即蚁群算法的正反馈原理存在不公平性竞争。为克服正反馈原理的弊端, 加快算法的收敛速度, 引入“奖励”和“惩罚”并存的机制, 其主要思想为:

首先对本次迭代搜索过的路径进行评价, 对优于上代解的本代解给与奖励, 额外增加此路径上的信息素, 以增加下一次循环迭代时蚂蚁选择此条路径的概率; 对劣于上代解的本代解给予惩罚, 减少该路径上的信息素量。“奖惩”机制的引入, 为搜索过程提供了奖惩分明的规则, 加快了算法的收敛速度。改进后的信息素全局更新方式如下:

$$\tau_{ij}^{new'} = \frac{L_{new}}{L_{old}} \times \tau_{ij}^{new} \quad (16)$$

由于将每条边的信息素水平限制在 $[\tau_{min}, \tau_{max}]$, 则:

$$\tau_{ij}^{new'} = \begin{cases} \tau_{max} & , \tau_{ij}^{new'} > \tau_{max} \\ \frac{\tau_{max} + \tau_{min}}{2} & , \tau_{ij}^{new'} < \tau_{min} \end{cases} \quad (17)$$

10) 判断是否等于最大迭代次数, 若满足则结束, 输出全局最优路径 L_{Global} , 否则清空访问表, NC++, 转 3), 重复进行上述步骤。

3 算法验证

设有 1 个雷达器材供应点和 9 个器材需求点, 各点坐标、各路段距离、通过时间、时间窗要求及各点需求量参数如表 1~表 5。

表 1 各点坐标表

点	坐标									
	V ₀	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	V ₈	V ₉
横坐标(km)	120	130	150	150	140	100	80	50	80	60
纵坐标(km)	185	160	165	190	210	210	190	190	180	160

表 2 各路段距离表 (km)

	V ₀	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	V ₈	V ₉
V ₀	0	26.93	30.06	30.41	32.02	32.02	40.31	70.18	40.31	65.00
V ₁		0	20.62	36.06	50.99	58.31	58.31	85.44	53.85	70.00
V ₂			0	25.00	46.10	67.27	74.33	103.01	71.59	90.14
V ₃				0	22.36	53.85	70.00	100.08	70.71	94.87
V ₄					0	40.00	63.25	92.20	67.08	94.34
V ₅						0	28.28	53.85	36.06	64.03
V ₆							0	30.00	10.00	36.06
V ₇								0	31.62	31.62
V ₈									0	28.28
V ₉										0

表 3 各路段通过所需时间表 (h)

	V ₀	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	V ₈	V ₉
V ₀	0	0.28	0.35	0.40	0.46	0.46	0.62	1.21	0.68	1.12
V ₁		0	0.22	0.58	0.84	0.98	0.95	1.49	0.90	1.17
V ₂			0	0.26	0.79	1.08	1.26	1.77	1.23	1.59
V ₃				0	0.26	0.86	1.14	1.69	1.18	1.72
V ₄					0	0.59	1.06	1.61	1.13	1.71
V ₅						0	0.31	0.89	0.67	1.03
V ₆							0	0.52	0.14	0.67
V ₇								0	0.56	0.52
V ₈									0	0.31
V ₉										0

表4 时间窗 (h)

点	时间窗								
	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	V ₈	V ₉
开始时间	0.2	0.2	0.1	0.3	0.4	0.6	1.1	0.3	0.6
终止时间	1.5	1.8	0.5	0.9	2.3	1.4	2.6	1.7	2.1

表5 各点需求量 (t)

点	V ₁	V ₂	V ₃	V ₄	V ₅	V ₆	V ₇	V ₈	V ₉
需求量	3.1	5.2	5.5	2.7	6.3	3.9	4.8	4.7	6.7

各参数设定为 $m=6, \sigma=3, \alpha=2, \beta=4, \rho=0.5, \lambda_1=0.6, \lambda_2=0.4, NC=150, Q=10$, 采用 MATLAB 编程语言对设计的优化算法进行编程, 随机运行得到结果见图 3, $L=489.842$ km, 需要 5 辆运输车, 运输路线分别为 0-2-1-0, 0-3-4-0, 0-5-0, 0-7-9-0, 0-8-6-0。使用传统蚁群算法求解得到的结果为 $L=578.720$ km, 需要 6 辆运输车, 运输路线分别为 0-3-0, 0-4-0, 0-6-0, 0-1-2-0, 0-7-5-0, 0-9-8-0。根据结果比较可证明, 使用设计的改进蚁群算法能够取得较好的计算结果, 设计的运输路程比传统蚁群算法设计的路程缩短 15.36%。

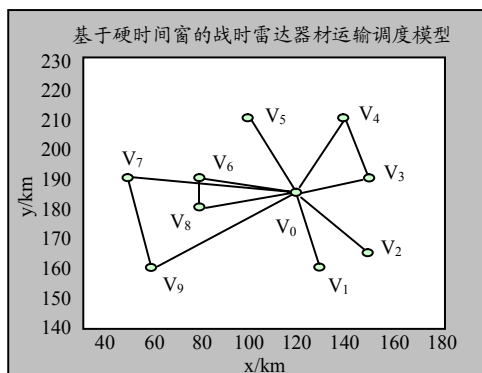


图3 基于硬时间窗的战时雷达器材运输调度路线图

4 结论

在综合考虑影响战时运输调度的指标因素的基础上建立了运输调度模型, 并针对基本蚁群算法容易早熟的问题, 设计一种改进的蚁群算法。经验证, 该算法简单、有效, 具有较好的实际操作意义。

参考文献:

[1] 段海滨. 蚁群算法原理及其应用[M]. 北京: 科学出版社, 2005.
 [2] 卢峰, 吴大舜, 曹兰. 蚁群算法理论及应用[J]. 科技广场, 2008, 15(5): 133-135.
 [3] 原思聪, 刘道华, 江祥奎, 等. 基于蚁群算法的多维有约束函数优化研究[J]. 计算机应用研究, 2008, 25(6): 1682-1684.
 [4] 朱庆保, 杨志军. 基于变异和动态信息素更新的蚁群优化算法[J]. 软件学报, 2004, 15(2): 185-192.

[5] 曹浪财, 罗健, 李天成. 智能蚂蚁算法-蚁群算法的改进[J]. 计算机应用研究, 2003, 22(10): 62-64.
 [6] 徐婕, 詹士昌. 动态调整信息素的蚁群算法[J]. 汉中师范学院学报, 2003, 21(2): 31-35.
 [7] 覃刚力, 杨家本. 自适应调整信息素的蚁群算法[J]. 信息与控制, 2002, 31(3): 31-35.
 [8] Mester,D,Brfiysy. Active guided evolution strategies for large scale vehicle routing problems with time windows[J]. Computers & Operations Research, 2005, 32(18): 1593-1614.
 [9] 高桂清, 王汉坤, 武建, 等. 基于模糊理论及蚁群算法打击任务量化限[J]. 四川兵工学报, 2009(10): 1-3.

 (上接第3页)

4 结束语

通过对 UML 的语义进行军事领域扩展, 将作战行动要素及关系纳入其中, 避免了在描述军事概念模型时产生语义混乱, 使其对军事概念模型的描述具有较好的针对性和适用性。实践结果表明: 运用扩展后的 UML 语法和语义, 从宏观和微观、从结构到行为的角度完整地抽象和描述了空中截击行动的整体过程、逻辑结构和行为动作, 所建立的模型能良好地转化为系统实现模型。将该方法应用于其它作战行动的概念建模, 也取得了良好的效果。

参考文献:

[1] 军事科学院. 中国人民解放军军语[S]. 北京: 军事科学出版社, 1997.
 [2] 王杏林, 曹晓东. 概念建模[M]. 北京: 国防工业出版社, 2007.
 [3] 张宏军. 军事建模理论与方法[M]. 南京: 解放军理工大学, 2003.
 [4] Grady Booch, 等. UML 用户指南[M]. 北京: 机械工业出版社, 2006.
 [5] 潘庆华. 基于 MDA 的军事仿真建模研究[D]. 南京: 解放军理工大学, 2009.
 [6] 胡晓峰, 等. 战争模拟引论[M]. 北京: 国防大学出版社, 2004.
 [7] 谭乐祖, 陈晓君, 李岩, 等. 坦克分队目标威胁评估与火力优化模型[J]. 四川兵工学报: 2009(1): 108-109.